7-1-2024

True Contraction Decomposition and Almost Eth-Tight Bipartization for Unit-Disk Graphs

Sayan Bandyapadhyay
Portland State University

William Lochet
Université de Montpellier

Daniel Lokshtanov
University of California, Santa Barbara

Saket Saurabh
Institute of Mathematical Sciences

Jie Xue
New York University Shanghai

Citation Details

Bandyapadhyay, S., Lochet, W., Lokshtanov, D., Saurabh, S., & Xue, J. (2024). True Contraction Decomposition and Almost ETH-Tight Bipartization for Unit-Disk Graphs. ACM Transactions on Algorithms, 20(3), 1–26.

True Contraction Decomposition and Almost ETH-Tight Bipartization for Unit-Disk Graphs

SAYAN BANDYAPADHYAY, Portland State University, Portland, Oregon, USA
WILLIAM LOCHET, LIRMM, Université de Montpellier, CNRS, Montpellier, France
DANIEL LOKSHTANOV, University of California, Santa Barbara, Santa Barbara, California, USA
SAKET SAURABH, Institute of Mathematical Sciences, Chennai, India
JIE XUE, New York University Shanghai, Shanghai, China

We prove a structural theorem for unit-disk graphs, which (roughly) states that given a set $\mathcal{D}$ of n unit disks inducing a unit-disk graph $G_{\mathcal{D}}$ and a number $p \in [n]$, one can partition $\mathcal{D}$ into p subsets $\mathcal{D}_1, \dots, \mathcal{D}_p$ such that for every $i \in [p]$ and every $\mathcal{D}' \subseteq \mathcal{D}_i$, the graph obtained from $G_{\mathcal{D}}$ by contracting all edges between the vertices in $\mathcal{D}_i \backslash \mathcal{D}'$ admits a tree decomposition in which each bag consists of $O(p + |\mathcal{D}'|)$ cliques. Our theorem can be viewed as an analog for unit-disk graphs of the structural theorems for planar graphs and almost-embeddable graphs proved recently by Marx et al. [SODA '22] and Bandyapadhyay et al. [SODA '22]. By applying our structural theorem, we give several new combinatorial and algorithmic results for unit-disk graphs. On the combinatorial side, we obtain the first Contraction Decomposition Theorem for unit-disk graphs, resolving an open question in the work by Panolan et al. [SODA '19]. On the algorithmic side, we obtain a new algorithm for bipartization (also known as odd cycle transversal) on unit-disk graphs, which runs in $2^{O(\sqrt{k} \log k)} \cdot n^{O(1)}$ time, where k denotes the solution size. Our algorithm significantly improves the previous slightly subexponential-time algorithm given by Lokshtanov et al. [SODA '22] which runs in $2^{O(k^{27/28})} \cdot n^{O(1)}$ time. We also show that the problem cannot be solved in $2^{o(\sqrt{k})} \cdot n^{O(1)}$ time assuming the Exponential Time Hypothesis, which implies that our algorithm is almost optimal.

CCS Concepts: • **Theory of computation** → **Computational geometry**; **Design and analysis of algorithms**;

Additional Key Words and Phrases: Unit-disk graphs, contraction decomposition, bipartization

ACM Reference format:
Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. 2024. True Contraction Decomposition and Almost ETH-Tight Bipartization for Unit-Disk Graphs. *ACM Trans. Algor.* 20, 3, Article 20 (June 2024), 26 pages.
https://doi.org/10.1145/3656042

A preliminary version of this paper appeared in the *Proceedings of the 38th International Symposium on Computational Geometry (SoCG)*, 2022.

Authors' Contact Information: Sayan Bandyapadhyay (Corresponding author), Portland State University, Portland, Oregon, USA; e-mail: sayanb@pdx.edu; William Lochet, LIRMM, Université de Montpellier, CNRS, Montpellier, France; e-mail: william.lochet@gmail.com; Daniel Lokshtanov, University of California, Santa Barbara, Santa Barbara, California, USA; e-mail: daniello@ucsb.edu; Saket Saurabh, Institute of Mathematical Sciences, Chennai, India; e-mail: saket@imsc.res.in; Jie Xue, New York University Shanghai, Shanghai, China; e-mail: jiexue@nyu.edu.

ACM 1549-6333/2024/6-ART20
https://doi.org/10.1145/3656042

1 Introduction

For a set $\mathcal{D}$ of unit disks in the plane, the unit-disk graph $G_{\mathcal{D}}$ induced by $\mathcal{D}$ has the unit disks in $\mathcal{D}$ as its vertices, where two vertices are connected by an edge whenever the two corresponding unit disks intersect. As one of the simplest but most important classes of geometric intersection graphs, unit-disk graphs have been extensively studied in various areas (e.g., computational geometry, graph theory, and algorithms) and find applications such as modeling the topology of *ad hoc* communication networks [30, 53]. The research on unit-disk graphs focused on both combinatorial aspects and algorithmic aspects.

In this article, we establish a structural theorem for unit-disk graphs, which leads to interesting new results in both combinatorial and algorithmic aspects. Our theorem can be viewed as a unit-disk-graph analog of the very recent theorems proved for planar graphs [42] and more generally for the so-called "almost-embeddable" graphs [6]. Thus, before introducing our theorem, let us first briefly review their results. Specifically, it was shown in [6, 42] that for a planar graph $G = (V, E)$ and a number $p \in [n]$ where $n = |V|$, one can partition V into $V_1, \ldots, V_p$ such that for every $i \in [p]$ and $V' \subseteq V_i$, the graph obtained from G by contracting all edges between the vertices in $V_i \backslash V'$ has treewidth $O(p + |V'|)$. Unfortunately, one can easily see that such a statement cannot hold for unit-disk graphs.[1] However, if we use the number of *cliques* (instead of vertices) in the bags of the tree decomposition to define its width, this statement is actually true for unit-disk graphs!

Let $\mathcal{D}$ be a set of n unit disks and $p \in [n]$ be any number. Our theorem (roughly) states that one can partition $\mathcal{D}$ into p subsets $\mathcal{D}_1, \ldots, \mathcal{D}_p$ such that for every $i \in [p]$ and every $\mathcal{D}' \subseteq \mathcal{D}_i$, the graph obtained from the unit-disk graph $G_{\mathcal{D}}$ by contracting all edges between the vertices in $\mathcal{D}_i \backslash \mathcal{D}'$ admits a tree decomposition in which each bag consists of $O(p + |\mathcal{D}'|)$ cliques. Furthermore, this partition can be computed in polynomial time. The formal statement of our theorem is more technical and will be presented in Theorem 3.1 after we introduce some preliminaries in Section 2. Note that the notion of tree decomposition with bags consisting of cliques is not new. In fact, this kind of tree decomposition has been widely applied on unit-disk graphs and other geometric intersection graphs to design efficient algorithms; see for example [13, 22, 47]. In what follows, we discuss the new combinatorial and algorithmic results derived from our main theorem.

Combinatorial Application: The First ***Contraction Decomposition Theorem (CDT)*** *on Unit-Disk Graphs.* In graph theory, a CDT is a statement of the following form: given a graph $G = (V, E)$ from some graph class, for any $p \in \mathbb{N}$, one can partition E into $E_1, \ldots, E_p$ such that contracting the edges in each E_i in G yields a graph of treewidth at most $f(p)$, for some function $f : \mathbb{N} \to \mathbb{N}$. CDT is classical tool useful in designing efficient approximation and parameterized algorithms in certain classes of graphs. Graph classes for which CDTs are known include planar graphs [34, 35], graphs of bounded genus [16], and H-minor-free graphs [15]. However, little was known about CDTs on geometric intersection graphs. Recently, Panolan et al. [48] made the first progress toward proving a CDT for unit-disk graphs. Specifically, they gave a weak version of CDT (which they call a *relaxed* CDT), in which the edge sets $E_1, \ldots, E_p$ need not to be disjoint; instead, it is required that each edge $e \in E$ is contained in $O(1)$ sets in $E_1, \ldots, E_p$. It remained open whether unit-disk graphs admit a "true" CDT (where $E_1, \ldots, E_p$ is a partition of E). In this article, by applying our main theorem, we give the first CDT for unit-disk graphs and hence resolve an open question of [48] (and also Hajiaghayi [29]). The function f in our CDT is quadratic, i.e., $f(p) = O(p^2)$, matching the bound in the weak CDT of [48].

[1] Indeed, the clique K_n is a unit-disk graph, and if we partition the vertices of K_n into p parts for $p \geq 2$, after contracting the smallest part, we get a clique of size at least $n/2$ which has treewidth $\Omega(n)$.

Algorithmic Application: Almost Exponential Time Hypothesis (ETH)-Tight Bipartization on Unit-Disk Graphs. Designing efficient algorithms on unit-disk graphs is a central topic in the study of unit-disk graphs. Many classical algorithmic problems have been studied on unit-disk graphs. Polynomial-time solvable problems include shortest paths [8, 9, 51], diameter computing [10, 25], maximum clique [11], and so forth. Compared to these problems, NP-hard problems received more attention on unit-disk graphs. In particular, studying parametrized algorithms [12] for these hard problems on unit-disk graphs (or other geometric intersection graphs) is one of the most active themes in recent years [2, 3, 21–24, 47] (also see the survey [48]). A well-known fact about parametrized complexity on planar graphs (or more generally, bounded-genus graphs and H-minor-free graphs) is the so-called "square root phenomenon:" many problems on planar graphs admit algorithms with running time $2^{\widetilde{O}(\sqrt{k})}n^{O(1)}$ or $n^{\widetilde{O}(\sqrt{k})}$, where k is the parameter (usually the solution size), and also admit (almost) matching lower bounds [7, 14, 17, 19, 20, 36, 37, 43, 45, 50]. Recently, it was shown that such a "square root phenomenon" also appears in many problems on unit-disk graphs. Specifically, algorithms with running time $2^{\widetilde{O}(\sqrt{k})}n^{O(1)}$ or $n^{\widetilde{O}(\sqrt{k})}$ were obtained on unit-disk graphs for Vertex Cover [13], Independent Set [3, 44], Feedback Vertex Set [4, 21], k-Path/Cycle [21, 23], and so forth and (almost) matching lower bounds were also known [13]. In this article, we apply our main theorem to add another classical problem to this family, namely, Bipartization.

In the Bipartization problem, one aims to make a graph bipartite by deleting as few vertices as possible. Formally, the input of Bipartization is a graph $G = (V, E)$ and a number k, and the goal is to determine whether there exists $X \subseteq V$ of size at most k such that $G - X$ is *bipartite*. In the literature, Bipartization is also called **Odd Cycle Transversal (OCT)**, as making a graph bipartite is equivalent to removing a set of vertices that hit all its odd cycles. As one of the most fundamental NP-complete problems in graph theory [52], Bipartization has been studied extensively over years [1, 18, 26, 31–33, 38, 49]. The best existing algorithm for Bipartization on general graphs runs in $2.3146^k n^{O(1)}$ time [39]. On planar graphs, a randomized algorithm with running time $2^{O(\sqrt{k}\log k)}n^{O(1)}$ was known [41, 42], and the same running time was achieved also for bounded-genus graphs and H-minor-free graphs very recently [6]. However, little was known about Bipartization on geometric intersection graphs. In fact, even achieving *slightly* subexponential-time parameterized algorithm for Bipartization on unit-disk graphs was a long-standing open problem, prior to the very recent work by Lokshtanov et al. [40]. The authors of [40] gave a randomized algorithm running in $2^{O(k^{\frac{27}{28}}\log k)}n^{O(1)}$ time for Bipartization on disk graphs (and thus unit-disk graphs), achieving the first $2^{o(k)}$ bound for the problem. This result, however, is still far away from showing the "square root phenomenon" for Bipartization on unit-disk graphs.

By applying our main theorem, we solve Bipartization on unit-disk graphs with a randomized algorithm running in $2^{O(\sqrt{k}\log k)}n^{O(1)}$ time, significantly improving the $2^{O(k^{\frac{27}{28}}\log k)}n^{O(1)}$ bound given by [40]. On the other hand, we establish an almost matching lower bound, showing that the problem cannot be solved in $2^{o(\sqrt{k})}n^{O(1)}$ time, assuming the ETH. Our results thus add Bipartization to the "square root" family of problems on unit-disk graphs. In terms of techniques, our algorithm solves the problem by first constructing the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of the unit-disk set $\mathcal{D}$ in our main theorem for $p = \sqrt{k}$ and then applying the well-known Baker's technique on $\mathcal{D}_1, \dots, \mathcal{D}_p$ together with a **Dynamic Programming (DP)** procedure similar to the one in [6] on tree decomposition. Such a scheme based on our theorem can possibly also be applied to solve other problems on unit-disk graphs. To give an example, we extend our algorithm to the problem of **Group Feedback Vertex Set (GFVS)** with non-identity labels, with the same running time.

Organization. The rest of the article is organized as follows. In Section 2, we introduce the basic notions and preliminaries used throughout the article. Our main theorem and its proof is given in Section 3, followed by its applications in Section 4. Finally, in Section 5, we conclude the article and raise some open questions for future study.

2 Preliminaries

The Canonical Grid. Consider the grid formed by vertical lines $\{x = i : i \in \mathbb{N}\}$ and horizontal lines $\{y = i : i \in \mathbb{N}\}$. We shall use it as the *canonical* grid throughout this article (in the rest of the article, we shall refer it as "the grid"). Each cell in the grid is a unit square, and we usually use the notation $\square$ to denote a cell. For a unit disk D, we denote by $\square_D$ the grid cell that contains the center of D. (For convenience, throughout the article, we always assume that the centers of the unit disks do *not* lie on the grid lines, and thus each center lies in exactly one cell of the grid. If this is not the case for the input unit disks, we can easily shift the grid or the unit disks to make the centers not lie on the grid lines.) For a set $\mathcal{D}$ of unit disks and a cell $\square$, we denote by $\mathcal{D} \Cap \square$ the subset of unit disks in $\mathcal{D}$ whose centers lie in $\square$. We say a subset $\mathcal{D}' \subseteq \mathcal{D}$ is *grid-respecting* if for any cell $\square$ such that $\mathcal{D}' \Cap \square \neq \emptyset$, we have $\mathcal{D}' \Cap \square = \mathcal{D} \Cap \square$. A partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of $\mathcal{D}$ is *grid-respecting* if $\mathcal{D}_1, \dots, \mathcal{D}_p$ are all grid-respecting subsets of $\mathcal{D}$.

Basic Graph Notions. Let $G = (V, E)$ be a graph. For a subset $V' \subseteq V$, the *subgraph* of G *induced* by V' is the graph consisting of the vertices in V' and the edges in E with both endpoints in V'. An *induced subgraph* of G is a subgraph of G induced by a subset of V. A vertex $v \in V$ is *neighboring* to a subset $V' \subseteq V$ in G if there exists $v' \in V'$ such that $(v, v') \in E$. A subset $V' \subseteq V$ is *neighboring* to another subset $V'' \subseteq V$ if there exist $v' \in V'$ and $v'' \in V''$ such that $(v', v'') \in E$.

Unit Disks and Unit-Disk Graphs. Let $\mathcal{D}$ be a set of unit disks in the plane, which are in general position in the sense that no two unit disks contact each other (i.e., intersect at a single point). For $D \in \mathcal{D}$, we denote by $\mathsf{ctr}(D)$ the *center* of the unit disk D. The union $U = \bigcup_{D \in \mathcal{D}} D$ is a closed region in the plane, whose boundary consists of a set of disjoint closed curves. The *outer boundary* of U is defined as the part of the boundary of U that is incident to the unbounded connected component of $\mathbb{R}^2 \backslash U$; see Figure 1 for an illustration. The *exposed* unit disks in $\mathcal{D}$ refers to the unit disks in $\mathcal{D}$ that intersect the outer boundary of U. In Figure 1, all unit disks in $\mathcal{D}$ are exposed. We denote by $\mathsf{Exp}(\mathcal{D})$ the set of exposed unit disks in $\mathcal{D}$. The *unit-disk graph* induced by $\mathcal{D}$, denoted by $G_{\mathcal{D}}$, has the unit disks in $\mathcal{D}$ as its vertices, where two vertices are connected by an edge whenever the two corresponding unit disks intersect.[2] We use $E_{\mathcal{D}}$ to denote the edge set of $G_{\mathcal{D}}$. Note that for a cell $\square$, the unit disks in $\mathcal{D} \Cap \square$ pairwise intersect and hence form a clique in $G_{\mathcal{D}}$, which we call a *cell clique*. We denote by $E^*_{\mathcal{D}} \subseteq E_{\mathcal{D}}$ the set of edges in all cell cliques in $G_{\mathcal{D}}$. For a subset $\mathcal{D}' \subseteq \mathcal{D}$, the unit-disk graph $G_{\mathcal{D}'}$ is canonically isomorphic to the subgraph of $G_{\mathcal{D}}$ induced by $\mathcal{D}'$. Thus, for convenience, we shall not distinguish between them: we shall also use $G_{\mathcal{D}'}$ to denote the induced subgraph of $G_{\mathcal{D}}$ and use $E_{\mathcal{D}'}$ to denote the set of edges in $G_{\mathcal{D}}$ between the vertices in $\mathcal{D}'$.

Tree Decomposition and Treewidth. With a bit abuse of notation, for a tree T, we also use T to denote the set of its nodes. A *tree decomposition* of a graph $G = (V, E)$ is a pair (T, β) where T is a tree and $\beta : T \to 2^V$ maps the nodes of T to subsets of V such that **(1)** $\bigcup_{t \in T} \beta(t) = V$, **(2)** for

[2]Without loss of generality, we can always assume that the unit disks defining a unit-disk graph are in general position. Indeed, one can convert a given set $\mathcal{D}_0$ of unit disks to another set $\mathcal{D}$ of unit disks in general position such that $G_{\mathcal{D}} = G_{\mathcal{D}_0}$. This is done as follows. First, we enlarge every unit disk in $\mathcal{D}_0$ to a disk of radius $1 + \varepsilon$, where ε is sufficiently small so that two disjoint unit disks in $\mathcal{D}_0$ are still disjoint after the enlargement. After this, we obtain a set $\mathcal{D}_1$ of congruent disks representing the same intersection graph as $\mathcal{D}_0$. Note that no two disks in $\mathcal{D}_1$ contact each other. Then by scaling we can further convert $\mathcal{D}_1$ to the desired set $\mathcal{D}$ of unit disks in general position.

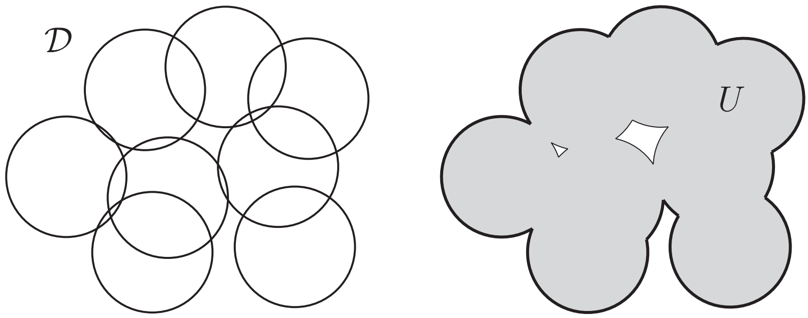

Fig. 1. The boundary and outer boundary of U (the heavier curve is the outer boundary).

each edge $(u, v) \in E$, there exists $t \in T$ with $u, v \in \beta(t)$, and **(3)** for each vertex $v \in V$, the nodes $t \in T$ with $v \in \beta(t)$ form a connected subset in T. Conventionally, we call $\beta(t)$ the *bag* of the node $t \in T$. The *width* of the tree decomposition (T, β) is $\max_{t \in T} |\beta(t)| - 1$. The *treewidth* of a graph G, denoted by $\mathbf{tw}(G)$ is the minimum width of a tree decomposition of G. It is sometimes more convenient to consider *rooted* trees. Thus, throughout this article, we always view the tree in a tree decomposition as a rooted tree. A tree decomposition (T, β) is *binary* if T is binary.

Lemma 2.1 (Chapter 7 in [12]). *Given an n-vertex graph G with $\mathbf{tw}(G) = w$, a binary tree decomposition of G of width $O(w)$ can be computed in $2^{O(w)} n^{O(1)}$ time.*

Edge Contraction. From a graph $G = (V, E)$, one can obtain a new graph via a so-called *edge contraction* operation. Specifically, by contracting an edge $e = (u, v) \in E$, we merge u and v into one vertex with edges connecting to both the neighbors of u and the neighbors of v in $V \backslash \{u, v\}$. More generally, we can contract a subset $E_0 \subseteq E$ of edges simply by contracting these edges "one-by-one." Formally, the resulting graph by contracting E_0 in G, which we denote by G/E_0, is defined as follows. The vertices of G/E_0 one-to-one corresponds to the connected components of the graph $G_0 = (V, E_0)$, and two vertices have an edge connecting them whenever the corresponding two connected components of G_0 are neighboring in G (i.e., there exists an edge in G whose two endpoints lie in the two components respectively). Let V_0 denote the vertex set of G/E_0. Associated to this edge contraction, there is a natural map $\pi : V \to V_0$ which maps each vertex $v \in V$ to the vertex of G/E_0 corresponding to the connected component of G_0 that contains v. We call π the *quotient map* of the edge contraction. Following is a well-known relation between tree decompositions of the graph after edge contraction and the original graph.

Fact 2.2. *Let $G = (V, E)$ be a graph obtained from another graph $G' = (V', E')$ via edge contraction with quotient map $\pi : V' \to V$. The following statements are true.*

(i) If (T, β) is a tree decomposition of G, then (T, β') is a tree decomposition of G' where $\beta'(t) = \pi^{-1}(\beta(t))$ for all nodes $t \in T$.
(ii) If (T', β') is a tree decomposition of G', then (T', β) is a tree decomposition of G where $\beta(t) = \pi(\beta'(t))$ for all nodes $t \in T'$.

Proof. To see (i), suppose (T, β) is a tree decomposition of G. As $\bigcup_{t \in T} \beta(t) = V$, we have $\bigcup_{t \in T} \beta'(t) = \bigcup_{t \in T} \pi^{-1}(\beta(t)) = V'$. Consider an edge $(u', v') \in E'$. If $\pi(u') = \pi(v') = v$, then any node $t \in T$ such that $v \in \beta(t)$ satisfies $u', v' \in \beta'(t)$; such a node exists as (T, β) is a tree decomposition of G. If $\pi(u') \neq \pi(v')$, then $(\pi(u'), \pi(v')) \in E$. In this case, there exists $t \in T$ such that $\pi(u'), \pi(v') \in \beta(t)$, which implies $u', v' \in \beta'(t)$. Finally, consider a vertex $v' \in V'$. The nodes

Algorithm 1 LAYERING($\mathcal{D}$) ▷ Output a layering $\ell : \mathcal{D} \to \mathbb{N}$

1: $q \leftarrow 0$
2: **while** $\mathcal{D} \neq \emptyset$ **do**
3: $\quad q \leftarrow q + 1$
4: $\quad \mathcal{X} \leftarrow \mathsf{Exp}(\mathcal{D})$
5: $\quad \mathcal{X}^+ = \bigcup_{X \in \mathcal{X}} (\mathcal{D} \Cap \Box_X)$
6: $\quad \mathsf{Tag}_X \leftarrow q$ for all $X \in \mathcal{X}^+$
7: $\quad \mathcal{D} \leftarrow \mathcal{D} \backslash \mathcal{X}^+$
8: **return** $\ell : D \mapsto \lceil \mathsf{Tag}_D / 100 \rceil$

$t \in T$ satisfying $\pi(v') \in \beta(t)$ are connected in T. These are exactly the nodes $t \in T$ satisfying $v' \in \beta'(t)$, and therefore they are connected in T. So (T, β') is a tree decomposition of G'.

To see (ii), suppose (T', β') is a tree decomposition of G'. As π is surjective and $\bigcup_{t \in T'} \beta'(t) = V'$, we have $\bigcup_{t \in T'} \beta(t) = \bigcup_{t \in T'} \pi(\beta'(t)) = V$. For an edge $(u, v) \in E$, there exist $u' \in \pi^{-1}(\{u\})$ and $v' \in \pi^{-1}(\{v\})$ such that $(u', v') \in E'$. Thus, $u', v' \in \beta'(t)$ for some node $t \in T'$. It follows that $u, v \in \beta(t)$. Finally, consider a vertex $v \in V$. For any node $t \in T'$, $v \in \beta(t)$ if and only if $\pi^{-1}(\{v\}) \cap \beta'(t) \neq \emptyset$. Note that $G'[\pi^{-1}(\{v\})]$ is a connected subgraph of G'. It is well-known that in a tree decomposition of a graph, the nodes whose bags intersect a connected subgraph are connected in the tree. This implies that the nodes $t \in T'$ such that $\pi^{-1}(\{v\}) \cap \beta'(t) \neq \emptyset$ are connected in T'. Therefore, the nodes $t \in T'$ satisfying $v \in \beta(t)$ are connected in T'. So (T', β) is a tree decomposition of G. □

3 The Main Theorem

In this section, we present the main theorem of this article, which establishes a structural property of unit-disk graphs. Formally, the theorem is the following.

THEOREM 3.1 (MAIN THEOREM). *Given a set $\mathcal{D}$ of n unit disks and an integer $p \in [n]$, one can compute in polynomial time a grid-respecting partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of $\mathcal{D}$ such that for every $i \in [p]$ and every $\mathcal{D}' \subseteq \mathcal{D}_i$, $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{D}'})) = O(p + |\mathcal{D}'|)$.*

Recall that in Section 1, we gave an informal version of the above theorem, which states that $G_{\mathcal{D}}/E_{\mathcal{D}_i \backslash \mathcal{D}'}$ admits a tree decomposition in which each bag contains $O(p + |\mathcal{D}'|)$ cliques. One may ask how Theorem 3.1 implies this statement. To see this, observe that $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{D}'})$ can be viewed as a graph obtained from $G_{\mathcal{D}}/E_{\mathcal{D}_i \backslash \mathcal{D}'}$ via edge contraction. Thus, if we start from a tree decomposition of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{D}'})$ of width $O(p + |\mathcal{D}'|)$ and apply Fact 2.2 to obtain a tree decomposition of $G_{\mathcal{D}}/E_{\mathcal{D}_i \backslash \mathcal{D}'}$, one can check that each bag of the latter tree decomposition consists of $O(p + |\mathcal{D}'|)$ cliques. We omit the details of this argument as it is not important. The rest of this section is dedicated to proving Theorem 3.1.

3.1 A Layering for the Unit Disks

The first step of proving Theorem 3.1 is to compute a *layering* for the unit disks in $\mathcal{D}$, that is, a decomposition of $\mathcal{D}$ into *layers*. We shall use a function $\ell : \mathcal{D} \to \mathbb{N}$ to represent the layering: the unit disks which are mapped to i by ℓ form the ith layer of $\mathcal{D}$. This layering ℓ *respects* the grid partition of $\mathcal{D}$ in the sense that $\ell^{-1}(\{i\})$ is a grid-respecting subset of $\mathcal{D}$ for all $i \in \mathbb{N}$. Besides, ℓ possesses some nice properties which will be used later to prove Theorem 3.1. Algorithm 1 presents the procedure for computing ℓ. In words, it iteratively finds the exposed unit disks in $\mathcal{D}$ (line 4) and removes from $\mathcal{D}$ the unit disks whose centers lie in the same cells as the centers of the exposed ones (lines 5 and 7), and finally combines the unit disks removed in every 100 iterations into one layer (line 8). Here the number 100 is arbitrarily chosen (any sufficiently large constant works).

It is clear that the layering ℓ returned by Algorithm 1 respects the cell partition of $\mathcal{D}$, because in line 6 we always assign the same tag to all unit disks with centers in the cells $\square_D$. We write $\mathcal{L}_i = \ell^{-1}(\{i\})$ and call it *the* ith *layer* of $\mathcal{D}$. Suppose there are in total m layers. We define $\mathcal{L}_{>i} = \bigcup_{j=i+1}^{m} \mathcal{L}_j$, $\mathcal{L}_{\geq i} = \bigcup_{j=i}^{m} \mathcal{L}_j$, $\mathcal{L}_{<i} = \bigcup_{j=1}^{i-1} \mathcal{L}_j$, $\mathcal{L}_{\leq i} = \bigcup_{j=1}^{i} \mathcal{L}_j$, and $\mathcal{L}_{[i,i']} = \bigcup_{j=i}^{i'} \mathcal{L}_j$. Next, we establish some nice properties of the layering ℓ.

Lemma 3.2. *The layering ℓ and the layers $\mathcal{L}_1, \ldots, \mathcal{L}_m$ satisfy the following three properties.*

(i) For any $D, D' \in \mathcal{D}$ such that $D \cap D' \neq \emptyset$, we have $|\ell(D) - \ell(D')| \leq 1$.
(ii) For a connected component of $G_{\mathcal{L}_{>i}}$ with vertex set $C \subseteq \mathcal{L}_{>i}$, the unit disks in $\mathcal{L}_i$ neighboring to C lie in the same connected component of $G_{\mathcal{L}_i}$.
(iii) For any $i, i' \in [m]$ with $i \leq i'$, $\mathbf{tw}\left(G_{\mathcal{L}_{[i,i']}} \middle/ E^*_{\mathcal{L}_{[i,i']}}\right) = O(i' - i + 1)$.

We remark that the construction of our layering ℓ on unit-disk graphs is analogous to (and also inspired by) the outerplanarity layering on planar graphs (which is obtained by iteratively removing the vertices on the boundary of the outer face of the planar graph). While for the outerplanarity layering the three properties in Lemma 3.2 follow easily, it requires considerably more work to show them for our layering on unit-disk graphs.

In the rest of this section, we prove Lemma 3.2. We begin with introducing some notations for ease of exposition. Since $\mathcal{D}$ changes during Algorithm 1, we denote by $\mathcal{D}^{(q)}$ the set $\mathcal{D}$ at the beginning of the qth iteration of the while-loop (lines 2–7). Define $\mathcal{X}^{(q)} = \mathsf{Exp}(\mathcal{D}^{(q)})$ and $U^{(q)}$ as the union of the unit disks in $\mathcal{D}^{(q)}$.

Verifying Property (i). Let $D, D' \in \mathcal{D}$ such that $D \cap D' \neq \emptyset$. To show $|\ell(D) - \ell(D')| \leq 1$, it suffices to show $|\mathsf{Tag}_D - \mathsf{Tag}_{D'}| \leq 100$. Let $q = \mathsf{Tag}_D$ and $q' = \mathsf{Tag}_{D'}$. If $q = q'$, we are done. If $q \neq q'$, we may assume $q < q'$ without loss of generality. Since $\mathsf{Tag}_D = q$, $D \in \mathcal{D} \Cap \square_X$ for some $X \in \mathcal{X}^{(q)}$. By the definition of $\mathcal{X}^{(q)}$, X intersects the outer boundary of $U^{(q)}$ and thus there exists a point $x \in X$ that is on the outer boundary of $U^{(q)}$. Let σ be the segment connecting x and $d' = \mathsf{ctr}(D')$. We say a cell $\square$ is *relevant* if there exists a unit disk in $\mathcal{D} \Cap \square$ that intersects σ. The following observation shows that the number of relevant cells is at least $q' - q + 1$.

Observation 3.3. *For each $i \in \{q, \ldots, q'\}$, there exists a unit disk $D_i \in \mathcal{D}$ with $\mathsf{Tag}_{D_i} = i$ that intersects σ. Thus, the number of relevant cells is at least $q' - q + 1$.*

Proof. Let $i \in \{q, \ldots, q'\}$. Note that $d' \in U^{(i)}$ as $D' \in \mathcal{D}^{(i)}$. On the other hand, x is either on or outside the outer boundary of $U^{(i)}$ (i.e., in the unbounded connected component of $\mathbb{R}^2 \backslash U^{(i)}$), because x is on the outer boundary of $U^{(q)}$ and $U^{(i)} \subseteq U^{(q)}$. As such, the segment σ should intersect the outer boundary of $U^{(i)}$. Consider the point a in the intersection of σ and the outer boundary of $U^{(i)}$. Since a is on the outer boundary of $U^{(i)}$, there exists a unit disk $D_i \in \mathcal{X}^{(i)}$ that contains a on its boundary. We have $\mathsf{Tag}_{D_i} = i$. Also, D_i intersects σ as $a \in D_i$. To bound the number of relevant cells, we notice that the cells $\square_{D_q}, \ldots, \square_{D_{q'}}$ are distinct, because the tags of $D_q, \ldots, D_{q'}$ are distinct. Furthermore, $\square_{D_q}, \ldots, \square_{D_{q'}}$ are all relevant cells, since $D_q, \ldots, D_{q'}$ intersect σ. So there are at least $q' - q + 1$ relevant cells. $\square$

Note that the length of σ is at most 3 because $D \cap D' \neq \emptyset$ and $D \cap X \neq \emptyset$. As such, there can be no more than 100 relevant cells (actually much fewer), because each relevant cell must contain a point with distance at most 1 from σ. Thus, $q' - q + 1 \leq 100$ and $|\ell(D) - \ell(D')| \leq 1$. Property (i) in Lemma 3.2 holds.

Verifying Property (ii). Consider a connected component of $G_{\mathcal{L}_{>i}}$ with vertex set $C \subseteq \mathcal{L}_{>i}$. Define $Q = \{q : \lceil q/100 \rceil = i\}$. For a fixed $q \in Q$, the outer boundary of $\mathcal{D}^{(q)}$ consists of some closed curves in the plane, each of which encloses a *region* that is topologically homeomorphic to a disk. These regions are clearly disjoint; we call the union of these regions the *domain* of $\mathcal{D}^{(q)}$. We claim that one of these regions should contain all unit disks in C. First, observe that the domain of $\mathcal{D}^{(q)}$ contains all unit disks in $\mathcal{D}^{(q)}$, and hence contains all disks in C since $C \subseteq \mathcal{L}_{>i} = \mathcal{D}^{(100i+1)} \subseteq \mathcal{D}^{(q)}$. Furthermore, because the regions are disjoint but G_C is connected, all unit disks in C must lie in the same region. We denote by R_q the region that contains the unit disks in C. We do this for all $q \in Q$, and thus obtain a set $\{R_q\}_{q \in Q}$ of regions. We observe that these regions are nested.

Observation 3.4. *$R_q \supseteq R_{q'}$ for all $q, q' \in Q$ with $q \leq q'$.*

Proof. Since $q \leq q'$, the domain of $\mathcal{D}^{(q)}$ contains the domain of $\mathcal{D}^{(q')}$ and in particular contains $R_{q'}$. Because $R_{q'}$ is connected, it is either contained in R_q or disjoint from R_q. As the unit disks in C are contained in both R_q and $R_{q'}$, we have $R_q \cap R'_q \neq \emptyset$ and thus $R_q \supseteq R_{q'}$. □

To prove property (ii), consider two unit disks $D, D' \in \mathcal{L}_i$ that are neighboring to C. Let $q = \mathsf{Tag}_D$ (resp., $q' = \mathsf{Tag}_{D'}$), then the tag of any unit disk in $\mathcal{D} \Cap \Box_D$ (resp., $\mathcal{D} \Cap \Box_{D'}$) is q (resp., q'). As $D, D' \in \mathcal{L}_i$, we have $q, q' \in Q$ and we assume $q \geq q'$ without loss of generality. Since D is neighboring to C and $\mathsf{Tag}_D = q$, D must be contained in R_q and thus all unit disks in $\mathcal{D} \Cap \Box_D$ are contained in R_q. Furthermore, there exists a unit disk $X \in \mathcal{D} \Cap \Box_D$ which is exposed in $\mathcal{D}^{(q)}$, i.e., $X \in \mathcal{X}^{(q)}$. Note that X must intersect the boundary of R_q, because X intersects the outer boundary of $U^{(q)}$ and is contained in R_q. Similarly, there exists a unit disk $X' \in \mathcal{D} \Cap \Box_{D'}$ exposed in $\mathcal{D}^{(q')}$ which intersects the boundary of $R_{q'}$.

Observation 3.5. *$D' \cup X'$ intersects the boundary of R_q.*

Proof. As X' intersects the boundary of $R_{q'}$, there exists a point $x' \in X'$ on the boundary of $R_{q'}$. Then either $x' \notin R_q$ or x' is on the boundary of R_q, because $R_q \subseteq R_{q'}$ by Observation 3.4. In the latter case, we are done, as X' intersects the boundary of R_q. So assume $x' \notin R_q$. Since D' is neighboring to C and the unit disks in C are all contained in R_q, we have $D' \cap R_q \neq \emptyset$. Now $D' \cup X'$ intersects R_q and contains a point x' that is outside R_q. Note that $D' \cup X'$ is connected, because $X' \in \mathcal{D} \Cap \Box_{D'}$. Therefore, $D' \cup X'$ intersects the boundary of R_q. □

Now both $D \cup X$ and $D' \cup X'$ are connected and intersect the boundary of R_q. Note that the unit disks in $\mathcal{D}^{(q)}$ that intersect the boundary of R_q form a connected unit-disk graph. Thus, the unit-disk graph induced by these unit disks together with D, X, D', X' is also connected. All these unit disks belong to $\mathcal{L}_i$, and are hence in the same connected component of $G_{\mathcal{L}_i}$. In particular, D and D' are in the same connected component of $G_{\mathcal{L}_i}$. Property (ii) in Lemma 3.2 holds.

Verifying Property (iii). We notice that, to verify property (iii), it suffices to show that $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}) = O(j)$ for all $j \in [m]$, because $\mathcal{L}_{[i,i']}$ is nothing but the first $j = i' - i + 1$ layers of the unit-disk set $\mathcal{L}_{\geq i}$. To this end, we first construct a drawing of the graph $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ on the plane (possibly with edge crossings). The vertices of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ one-to-one correspond to the cells $\Box$ for which $\mathcal{L}_{\leq j} \Cap \Box \neq \emptyset$, and we denote by $v(\Box)$ the vertex corresponding to the cell $\Box$. We draw each vertex $v(\Box)$ at an arbitrary point inside the cell $\Box$ that lies in the intersection of all unit disks in $\mathcal{D} \Cap \Box$ (such a point always exists, e.g., the center of $\Box$). For simplicity, we also use $v(\Box)$ to denote the point in the plane where we draw the vertex $v(\Box)$. For each edge $e = (v(\Box), v(\Box'))$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, we draw it as a polyline (or polygonal chain) in the plane connecting $v(\Box)$ and $v(\Box')$ as follows. Since $v(\Box)$ and $v(\Box')$ are connected by an edge in $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, there exist unit disks

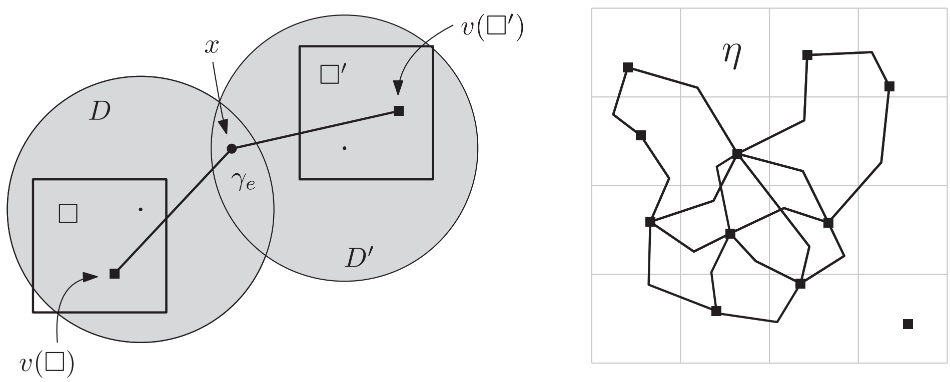

Fig. 2. Illustrating the drawing η. The left part is the construction of one edge curve γ_e and the right part is an example of how the drawing η finally looks like.

$D \in \mathcal{L}_{\leq j} \Cap \square$ and $D' \in \mathcal{L}_{\leq j} \Cap \square'$ such that $D \cap D' \neq \emptyset$. We choose an arbitrary point $x \in D \cap D'$ and let σ be the segment connecting x and $v(\square)$, and σ' be the segment connecting x and $v(\square')$. We then draw the edge e as the two-piece polyline consisting of the segments σ and σ', and denote this polyline by γ_e. See the left part of Figure 2 for an illustration. Note that γ_e is contained in $D \cup D'$. In this way, we obtain a plane drawing of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ (possibly with edge crossings), and denote this drawing by η. For convenience, we call the polylines γ_e *edge curves*. By carefully choosing the middle points of the edge curves, we can guarantee that all segments of the edge curves have different slopes (and thus two edge curves can only intersect at finitely many points) and no three edge curves intersect at a common point. It is easy to see that each edge curve only intersects a constant number of other edge curves, and thus η embeds $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ in the plane with $O(1)$ crossings per edge. Grigoriev and Bodlaender [27] showed that the treewidth of such a graph is linear in its diameter. Unfortunately, we cannot directly apply this result to bound the treewidth of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, because the diameter of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ might be unbounded. However, the ideas in the proof of [27] turn out to be useful in our setting as well. We shall use an argument similar to that in [27]: constructing a planar graph P from the drawing γ by adding vertices to the edge-crossing points and then bounding $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}})$ by considering $\mathbf{tw}(P)$. To do this, we first observe some basic properties of the drawing η.

Let Γ be the image of η in the plane, which is equal to the union of all edge curves and all $v(\square)$; see the right part of Figure 2. By our construction, Γ is contained in the union of all unit disks in $\mathcal{D}$. Next, we establish some properties of Γ, which will be used later for bounding $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}})$. For two points $a, b \in \mathbb{R}^2$, a *path* from a to b is a continuous map $f : [0,1] \to \mathbb{R}^2$ with $f(0) = a$ and $f(1) = b$. We write $\Delta(f, \Gamma) = |\{x \in [0,1] : f(x) \in \Gamma\}|$; if $\{x \in [0,1] : f(x) \in \Gamma\}$ is not finite, we simply set $\Delta(f, \Gamma) = \infty$.

Observation 3.6. *For any two points $a, b \in \mathbb{R}^2$ with distance d, there exists a path $f : [0,1] \to \mathbb{R}^2$ from a to b such that $\Delta(f, \Gamma) = O(d)$.*

Proof. Pick an arbitrary point c which has distance at most d to both a and b and satisfies that the slopes of the segments ac and cb are different from the slopes of all segments in the edge curves. Define $f : [0,1] \to \mathbb{R}^2$ as the path from a to b which first goes from a to c along with the segment ac and then goes from c to b along with the segment cb. Since the slope of ac (resp., cb) is different from the slopes of the segments in the edge curves, each edge curve can intersect ac (resp., cb) at (at most) two points. Therefore, $\Delta(f, \Gamma)$ is finite. To show $\Delta(f, \Gamma) = O(d)$, it suffices to show that the segment ac (resp., cb) only intersects $O(d)$ edge curves. Without loss of generality, we only consider the segment ac. Let $e = (v(\square), v(\square'))$ be an edge whose edge curve γ_e intersects ac. We claim that the distance from any point in $\square$ (resp., $\square'$) to ac is $O(1)$. Indeed, by our construction, the edge curve γ_e consists of two segments of length at most 2, and the two endpoints of γ_e lie in $\square$ and $\square'$, respectively. Thus, the distance between a point in $\square$ (resp., $\square'$) and an intersection point of γ_e and ac cannot be larger than $4 + \sqrt{2}$, where $\sqrt{2}$ is the maximum distance between two points in $\square$ (resp., $\square'$). It follows that the distance from any point in $\square$ (resp., $\square'$) to ac is at most $4 + \sqrt{2}$. Based on this observation, we see that the edges of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ whose edge curves intersect ac must be between the cells with constant distance from ac. Since the length of ac is at most d, there can be only $O(d)$ cells with constant distance from ac. Thus, ac intersects $O(d)$ edge curves. □

Observation 3.7. *For any point $a \in \mathbb{R}^2$, there exists a point b in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$ and a path $f : [0, 1] \to \mathbb{R}^2$ from a to b such that $\Delta(f, \Gamma) = O(j)$.*

Proof. We first consider a special case where $a = v(\square)$ for some vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$. We show that if the unit disks in $\mathcal{D} \Cap \square$ have tag q, then there exists a point b in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$ and a path $f : [0, 1] \to \mathbb{R}^2$ from $v(\square)$ to b such that $\Delta(f, \Gamma) = O(q)$. We use induction on q. The base case is $q = 1$. If the tag of the unit disks in $\mathcal{D} \Cap \square$ is 1, then there exists a unit disk in $\mathcal{D} \Cap \square$ that is exposed in $\mathcal{D}$. This implies that $\square$ is “close” to the outer boundary of $U = \bigcup_{D \in \mathcal{D}} D$; more precisely, one can find a point b in the unbounded connected component of $\mathbb{R}^2 \backslash U$ such that the distance between $v(\square)$ and b is $O(1)$. Recall that $\Gamma \subseteq U$, and so b lies in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$. By Observation 3.6, there exists a path f from $v(\square)$ to b such that $\Delta(f, \Gamma) = O(1)$. Now assume the statement holds for all $q \in \{1, \ldots, k-1\}$. Consider the case $q = k$, i.e. the tag of the unit disks in $\mathcal{D} \Cap \square$ is k. There exists a unit disk in $\mathcal{D} \Cap \square$ which is exposed in $\mathcal{D}^{(k)}$, which implies the existence of a point on the outer boundary of $U^{(k)}$ with distance $O(1)$ from $v(\square)$. As such, there also exists a point b_1 with distance $O(1)$ from $v(\square)$ that is outside the outer boundary of $U^{(k)}$, i.e., in the unbounded connected component of $\mathbb{R}^2 \backslash U^{(k)}$. We distinguish two cases: $b_1 \in U$ and $b_1 \notin U$.

If $b_1 \in U$, then $b_1 \in U \backslash U^{(k)}$. Thus, there must exist a unit disk $D \in \mathcal{D} \backslash \mathcal{D}^{(k)}$ that contains b_1. Let $\square' = \square_D$. The distance between $v(\square)$ and $v(\square')$ is $O(1)$, because b_1 is with distance $O(1)$ from $v(\square)$ and b_1 lies in $D \in \mathcal{D} \Cap \square'$. By Observation 3.6, there exists a path f_1 from $v(\square)$ to $v(\square')$ such that $\Delta(f_1, \Gamma) = O(1)$. On the other hand, the tag of the unit disks in $\mathcal{D} \Cap \square'$ is $q = \mathrm{Tag}_D \in \{1, \ldots, k-1\}$. So by our induction hypothesis, there exists a path f_2 from $v(\square')$ to a point b in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$ such that $\Delta(f_2, \Gamma) = O(q)$. By concatenating f_1 and f_2, we obtain a path $f : [0, 1] \to \mathbb{R}^2$ from $v(\square)$ to b such that $\Delta(f, \Gamma) = \Delta(f_1, \Gamma) + \Delta(f_2, \Gamma) = O(k)$.

Now consider the other case where $b_1 \notin U$. If b_1 is in the unbounded connected component of $\mathbb{R}^2 \backslash U$, then b_1 is in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$. In this case, we can simply set $b = b_1$ and by Observation 3.6 there exists a path $f : [0, 1] \to \mathbb{R}^2$ from $v(\square)$ to b such that $\Delta(f, \Gamma) = O(1)$. So it suffices to consider the case where $b_1 \in C$ for some bounded connected component C of $\mathbb{R}^2 \backslash U$. We have $U^{(k)} \subseteq U$ and thus $C \subseteq \mathbb{R}^2 \backslash U^{(k)}$. Also, because b_1 lies in the unbounded connected component of $\mathbb{R}^2 \backslash U^{(k)}$, C is also contained in the unbounded connected component of $\mathbb{R}^2 \backslash U^{(k)}$. It follows that the boundary of the closure of C is contained in U but not contained in $U^{(k)}$. In particular, we can find a point b_2 on the boundary of the closure of C such that $b_2 \in U \backslash U^{(k)}$. Then there exists a unit disk $D \in \mathcal{D} \backslash \mathcal{D}^{(k)}$ that contains b_2. Let $\square' = \square_D$. Note that the distance between $v(\square)$ and b_1 is $O(1)$, and the distance between b_2 and $v(\square')$ is also $O(1)$. Thus, by Observation 3.6, there exist a path g_1 from $v(\square)$ to b_1 and a path g_2 from b_2 to $v(\square')$ such that $\Delta(g_1, \Gamma) = O(1)$ and $\Delta(g_2, \Gamma) = O(1)$. Furthermore, there exists a path $g : [0, 1] \to \mathbb{R}^2$ from b_1 to b_2 such that $g(x) \in C$ for all $0 \leq x < 1$, because $b_1 \in C$, b_2 is on the boundary of the closure of

C, and C is connected. Since Γ is contained in U, we have $\Gamma \cap C = \emptyset$, which implies $\Delta(g, \Gamma) \leq 1$. By concatenating g_1, g, g_2, we obtain a path f_1 from $v(\square)$ to $v(\square')$ such that $\Delta(f_1, \Gamma) = O(1)$. On the other hand, the tag of $\square'$ is $q = \mathsf{Tag}_D \in \{1, \dots, k-1\}$. By our induction hypothesis, there exists a path f_2 from $v(\square')$ to a point b in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$ such that $\Delta(f_2, \Gamma) = O(q)$. Finally, by concatenating f_1 and f_2, we obtain a path $f : [0, 1] \to \mathbb{R}^2$ from $v(\square)$ to b such that $\Delta(f, \Gamma) = \Delta(f_1, \Gamma) + \Delta(f_2, \Gamma) = O(k)$. This completes our induction argument and shows that for a vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, there exists a point b in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$ and a path f from $v(\square)$ to b such that $\Delta(f, \Gamma) = O(q)$, where q is the tag of the unit disks in $\mathcal{D} \Cap \square$.

Note that for any vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, the tag of the unit disks in $\mathcal{D} \Cap \square$ is at most $100j$, and is hence $O(j)$. Thus, so far we have proved the statement in the observation for the special case where $a = v(\square)$ for some vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$. To prove for the general case where a is an arbitrary point in $\mathbb{R}^2$, we observe that there always exists a path g from a to some vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ such that $\Delta(g, \Gamma) = O(1)$. If $a \in \Gamma$, then a is on some edge curve γ_e. In this case, a is within distance $O(1)$ from an endpoint $v(\square)$ of e and thus by Observation 3.6, there exists a path g from a to $v(\square)$ such that $\Delta(g, \Gamma) = O(1)$. If $a \notin \Gamma$, then a lies in some connected component C of $\mathbb{R}^2 \backslash \Gamma$. Pick a point a' on the boundary of the closure of C. Then a' is on some edge curve γ_e and thus there exists a path g_2 from a' to some vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ such that $\Delta(g_2, \Gamma) = O(1)$. Also, because of the choice of a', there exists a path g_1 from a to a' such that $g_1(x) \in C$ for all $0 \leq x < 1$ and thus $\Delta(g_1, \Gamma) = 1$. By concatenating g_1 and g_2, we obtain a path g from a to $v(\square)$ such that $\Delta(g, \Gamma) = O(1)$. This directly completes the proof. Indeed, for any $a \in \mathbb{R}^2$, there exists a path g from a to some vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ such that $\Delta(g, \Gamma) = O(1)$, and as argued before there exists a path g' from $v(\square)$ to some point b in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$ such that $\Delta(g', \Gamma) = O(j)$. By concatenating g and g', we obtain a path f from a to b such that $\Delta(f, \Gamma) = O(j)$. □

The plane drawing η of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ naturally induces a planar graph P as follows. The vertex set of P consists of the vertices of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ and the edge-crossing points in the drawing η (called *crossings* for short). Two vertices of P are connected by an edge if they are "adjacent" on some edge curve γ_e. Formally, consider an edge $e = (v(\square), v(\square'))$ of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$. Suppose the crossings on γ_e are $c_1, \dots, c_r$, ordered from the $v(\square)$ end to the $v(\square')$ end. Then we include in P the edges $(v(\square), c_1), (c_1, c_2), \dots, (c_{r-1}, c_r), (c_r, v(\square'))$. After considering all edges of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, we complete the construction of P. Note that η naturally induces a planar drawing of P (thus P is planar), which we denote by η_0. Clearly, the image of η_0 is equal to the image of η, which is Γ. See Figure 3 for an illustration of the construction of P. The following observation gives a relation between the treewidths of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ and P.

Observation 3.8. $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}) \leq O(\mathbf{tw}(P))$.

Proof. For each vertex v of P, we define its *witness set* $\mathsf{wit}(v)$ as a set of vertices of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ as follows. If v itself is a vertex of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, we simply define $\mathsf{wit}(v) = \{v\}$. If v is a crossing of the drawing η, then it is contributed by two edges of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, and we let $\mathsf{wit}(v)$ consist of the four vertices of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ incident to these two edges. Now consider a tree decomposition (T, β) of P. Define $\beta^*(t) = \bigcup_{v \in \beta(t)} \mathsf{wit}(v)$ for all nodes $t \in T$. We claim that (T, β^*) is a tree decomposition of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ of width $O(\mathbf{tw}(P))$. Note that $|\beta^*(t)| \leq 4|\beta(t)|$ for all $t \in T$ as the witness set of every vertex of P is of size at most 4. Thus, the width of (T, β^*) is $O(\mathbf{tw}(P))$ and it suffices to show that

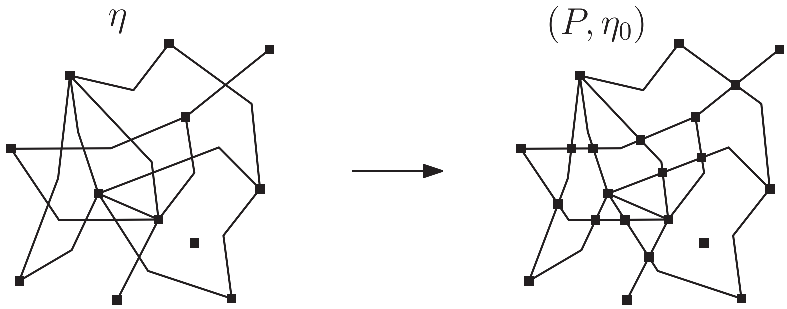

Fig. 3. Illustrating the planar graph P obtained by adding vertices to the crossings of η.

(T, β^*) is a tree decomposition of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$. First, each vertex v of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ is also a vertex of P, so there exists a node $t \in T$ with $v \in \beta(t)$, which implies $v \in \beta^*(t)$. Second, we show that for each edge (u, v) of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$, there exists $t \in T$ such that $u, v \in \beta^*(t)$. If there is no crossing on the image of (u, v) under η, then (u, v) is also an edge in P. Since (T, β) is a tree decomposition of P, there exists $t \in T$ such that $u, v \in \beta(t)$ and hence $u, v \in \beta^*(t)$. If there is a crossing x on the image of (u, v), then $u, v \in \text{wit}(x)$. In this case, we have $u, v \in \beta^*(t)$ for any node $t \in T$ such that $x \in \beta(t)$. Finally, it suffices to verify that for each vertex v of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ the nodes $t \in T$ with $v \in \beta^*(t)$ are connected in T. Let E_v be the set of edges of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ incident to v, and X_v be the set of vertices of P whose witness sets contain v. Observe that X_v consists of v itself and all crossings on the images of the edges in E_v under η. Also, X_v is connected in P. It is well-known that in a tree decomposition of a graph, the nodes whose bags intersect a connected subgraph are connected in the tree. Therefore, the nodes $t \in T$ satisfying $X_v \cap \beta(t) \neq \emptyset$ are connected in T. Note that $v \in \beta^*(t)$ if and only if $X_v \cap \beta(t) \neq \emptyset$ for all $t \in T$. So the nodes $t \in T$ satisfying $v \in \beta^*(t)$ are connected in T. It follows that (T, β^*) is a tree decomposition of $G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}$ of width $O(\mathbf{tw}(P))$, and thus $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}) \leq O(\mathbf{tw}(P))$. □

Based on the above observation, it now suffices to show that $\mathbf{tw}(P) = O(j)$. To this end, we need to introduce a notion called *vertex-face incidence graph*. We consider the plane-embedded graph (P, η_0). The *vertex-face incidence graph* P^+ of (P, η_0) is a bipartite graph defined as follows. One part of P^+ consists of the vertices of (P, η_0), while the other part consists of the faces of (P, η_0). A vertex v of (P, η_0) and a face F of (P, η_0) are connected by an edge in P^+ if v is incident to F. Let o be the outer face of (P, η_0), which is a vertex of P^+. The *depth* of a vertex v in (P, η_0) is defined as the shortest-path distance between o and v in P^+. It is well-known that $\mathbf{tw}(P)$ is linear in the maximum depth of a vertex in (P, η_0); see for example [6] (Lemma 11 in the arxiv version). So we only need to show the depth of each vertex in (P, η_0) is $O(j)$.

Consider a vertex v of (P, η_0). By Observation 3.7, there exists a point b in the unbounded connected component of $\mathbb{R}^2 \backslash \Gamma$ and a path $f : [0, 1] \to \mathbb{R}^2$ from v to b such that $\Delta(f, \Gamma) = O(j)$. Suppose $\{x \in [0, 1] : f(x) \in \Gamma\} = \{x_1, \ldots, x_k\}$ where $k = O(j)$ and $x_1 < \cdots < x_k$. We have $x_1 = 0$ because $f(0) = v \in \Gamma$. Let $I_i = \{x : x_i < x < x_{i+1}\}$ for $i \in [k-1]$ and $I_k = \{x : x_k < x \leq 1\}$. Since f is continuous, the image of each I_i under f is connected and disjoint from Γ, and hence lies in one face of (P, η_0), which we denote by F_i. We say two faces of (P, η_0) are *adjacent* if they are incident to a common vertex of (P, η_0). Clearly, the shortest-path distance between two adjacent faces of (P, η_0) in P^+ is 2. Note that for each $i \in [k-1]$, F_i and F_{i+1} are adjacent, as they are both incident to the point $f(x_{i+1}) \in \Gamma$, which is either a vertex of (P, η_0) or on an edge e of (P, η_0); in the latter case, F_i and F_{i+1} are both incident to the two endpoints of e. Therefore, the shortest-path distance between

F_1 and F_k in P^+ is at most $2k-2$, which is $O(j)$. Now F_1 is incident to $f(x_1) = f(0) = v$ and F_k is the outer face o of (P, η_0) since $b \in F_k$. It follows that the shortest-path distance between v and o is $O(j)$, and thus the depth of v is $O(j)$. This implies $\mathbf{tw}(P) = O(j)$ and hence $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E^*_{\mathcal{L}_{\leq j}}) = O(j)$ by Observation 3.8. Property (iii) in Lemma 3.2 holds.

3.2 Constructing the Partition $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$

Given the layering ℓ of $\mathcal{D}$ presented in the previous section, we are able to construct the partition $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$ of $\mathcal{D}$ in Theorem 3.1. The basic idea is similar to that used in Baker's technique: combining the congruent layers modulo p. Recall that $\mathcal{L}_1, \ldots, \mathcal{L}_m$ are the layers of $\mathcal{D}$. We define $\mathcal{D}_i = \bigcup_{j=0}^{\lfloor (m-i)/p \rfloor} \mathcal{L}_{jp+i}$, i.e., $\mathcal{D}_i$ consists of all layers whose index is congruent to i modulo p. Clearly, $\mathcal{D}_1, \ldots, \mathcal{D}_p$ can be computed in polynomial time. As $\{\mathcal{L}_1, \ldots, \mathcal{L}_m\}$ is a partition of $\mathcal{D}$, $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$ is also a partition of $\mathcal{D}$. Also, since each $\mathcal{L}_i$ is a grid-respecting subset of $\mathcal{D}$, the partition $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$ of $\mathcal{D}$ is grid-respecting. To prove Theorem 3.1, it suffices to show $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{D}'})) = O(p + |\mathcal{D}'|)$ for any $i \in [p]$ and $\mathcal{D}' \subseteq \mathcal{D}_i$.

3.3 Bounding the Treewidth When $\mathcal{D}' = \emptyset$

In this section, we prove a special case of the treewidth bound in Theorem 3.1 where $\mathcal{D}' = \emptyset$. In other words, we show $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})) = O(p)$ for any $i \in [p]$. If $p = 1$, we are done, as in this case $\mathcal{D}_1 = \mathcal{D}$ and $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_1})$ is a graph without edges, which has treewidth 0. So assume $p \geq 2$. Our proof for all $i \in [p]$ is identical, so we only consider the case where $i = p$, i.e., we show $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})) = O(p)$. For convenience, we set $\mathcal{L}_i = \emptyset$ for all $i \leq 0$ and $i > m$. Define $r = \lfloor m/p \rfloor + 1$ and $i_j = (j-1) \cdot p$ for $j \in \mathbb{N}$. So we have $\mathcal{D}_p = \bigcup_{j=1}^{r} \mathcal{L}_{i_j}$.

To bound the treewidth of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$, we first define a support tree T_{supp} as follows. Roughly speaking, T_{supp} is a tree that interprets the containment relation between the connected components of $G_{\mathcal{L}_{>i_1}}, \ldots, G_{\mathcal{L}_{>i_r}}$. The depth of T_{supp} is r. The root (i.e., the node at the 0th level) of T_{supp} is a dummy node. For all $j \in [r]$, the nodes at the jth level of T_{supp} are one-to-one corresponding to the connected components of $G_{\mathcal{L}_{>i_j}}$. The parent of the nodes at the first level is just the root. The parents of the nodes at the lower levels are defined as follows. Consider a node $t \in T_{\text{supp}}$ at the jth level for $j \geq 2$. Since $G_{\mathcal{L}_{>i_j}}$ is a subgraph of $G_{\mathcal{L}_{>i_{j-1}}}$, the connected component of $G_{\mathcal{L}_{>i_j}}$ corresponding to t is contained in a unique connected component of $G_{\mathcal{L}_{>i_{j-1}}}$, which corresponds to a node t' at the $(j-1)$-th level of T_{supp}. We then define the parent of t as t'. For each node $t \in T_{\text{supp}}$, we associate to t a set $\mathcal{A}_t \subseteq \mathcal{D}$ defined as follows. If t is the root, $\mathcal{A}_t = \emptyset$. Suppose t is at the jth level for $j \in [r]$ and let $C_t \subseteq \mathcal{L}_{>i_j}$ be the vertex set of the connected component of $G_{\mathcal{L}_{>i_j}}$ corresponding to t. Then we define $\mathcal{A}_t = \{D \in C_t : i_j < \ell(D) \leq i_{j+1}\}$, i.e., $\mathcal{A}_t$ consists of all unit disks in C_t which lie in the layers $\mathcal{L}_{i_j+1}, \ldots, \mathcal{L}_{i_{j+1}}$.

Observation 3.9. *$\{\mathcal{A}_t\}_{t \in T_{\text{supp}}}$ is a grid-respecting partition of $\mathcal{D}$. Furthermore, the vertices of each connected component of $G_{\mathcal{D}_p}$ are contained in the same $\mathcal{A}_t$.*

Proof. We first observe that every $D \in \mathcal{D}$ belongs to $\mathcal{A}_t$ for some $t \in T_{\text{supp}}$. Indeed, there exists some $j \in [r]$ such that $i_j < \ell(D) \leq i_{j+1}$. Then $D \in \mathcal{L}_{>i_j}$ and thus D is contained in some connected component of $G_{\mathcal{L}_{>i_j}}$, which corresponds to a node $t \in T_{\text{supp}}$ at the jth level of T_{supp}. By definition, we have $D \in \mathcal{A}_t$. Next, we observe that $\mathcal{A}_t \cap \mathcal{A}_{t'} = \emptyset$ for different nodes $t, t' \in T_{\text{supp}}$. Suppose t (resp., t') is at the jth (resp., j'th) level. If $j < j'$, then $\mathcal{A}_t \cap \mathcal{A}_{t'} = \emptyset$, as $\ell(D) \leq i_{j+1} \leq i_{j'} < \ell(D')$ for all $D \in \mathcal{A}_t$ and $D' \in \mathcal{A}_{t'}$. Similarly, we have $\mathcal{A}_t \cap \mathcal{A}_{t'} = \emptyset$ if $j > j'$. So it suffices to consider the case $j = j'$. In this case, since $t \neq t'$, t and t' correspond to different connected components of $G_{\mathcal{L}_{>i_j}}$ which contain the vertices in $\mathcal{A}_t$ and $\mathcal{A}_{t'}$ respectively. Hence, $\mathcal{A}_t \cap \mathcal{A}_{t'} = \emptyset$. This shows that $\{\mathcal{A}_t\}_{t \in T_{\text{supp}}}$ is a partition of $\mathcal{D}$. To see that this partition is grid-respecting, consider a unit disk

$D \in \mathcal{D}$. Suppose $D \in \mathcal{A}_t$ for a node $t \in T_{\mathsf{supp}}$ at the jth level. Then all unit disks in $\mathcal{D} \Cap \square_D$ are contained in $\mathcal{A}_t$ because they are in the same layer and belong to the same connected component of $G_{\mathcal{L}_{>i_j}}$.

To show the second statement, recall that $\mathcal{D}_p = \bigcup_{j=1}^r \mathcal{L}_{i_j}$. As we assumed $p \geq 2$, by property (i) of Lemma 3.2, the layers $\mathcal{L}_{i_1}, \dots, \mathcal{L}_{i_r}$ are pairwise non-adjacent. Therefore, the vertices of a connected component of $G_{\mathcal{D}_p}$ must lie in the same layer $\mathcal{L}_{i_j}$ for some $j \in [r]$. These vertices are thus contained in the same connected component of $G_{\mathcal{L}_{>i_{j-1}}}$ (as their corresponding unit disks form a connected unit-disk graph), and hence contained in the same $\mathcal{A}_t$ for some $t \in T_{\mathsf{supp}}$ at the $(j-1)$-th level. □

Observation 3.10. *Let $D \in \mathcal{A}_t$ and $D' \in \mathcal{A}_{t'}$ for different nodes $t, t' \in T_{\mathsf{supp}}$. If $D \cap D' \neq \emptyset$, then either t is the parent of t' or t' is the parent of t.*

Proof. Suppose t (resp., t') is at the jth (resp., j'th) level. By property (i) of Lemma 3.2, we have $|\ell(D) - \ell(D')| \leq 1$, which implies $|j - j'| \leq 1$. If $j = j'$, then $\mathcal{A}_t$ and $\mathcal{A}_{t'}$ belong to different connected components of $G_{\mathcal{L}_{>i_j}}$, which contradicts the fact $D \cap D' \neq \emptyset$. So we have either $j = j' + 1$ or $j' = j + 1$. Without loss of generality, assume $j = j' + 1$. Let $t^* \in T_{\mathsf{supp}}$ be the parent of t, and we claim that $t^* = t'$. Indeed, both t^* and t' are at the j'th level of T_{supp}. If $t^* \neq t'$, then t^* and t' correspond to two different connected components of $G_{\mathcal{L}_{>i_{j'}}}$, which contain D and D' respectively. This contradicts the fact $D \cap D' \neq \emptyset$. Thus $t^* = t'$. □

For each $t \in T_{\mathsf{supp}}$, we define a graph $J_t = G_{\mathcal{A}_t}/(E^*_{\mathcal{A}_t} \cup E_{\mathcal{A}_t \cap \mathcal{D}_p})$. Using Observation 3.9, one can easily verify that the edges in $E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p}$ that are incident to $\mathcal{A}_t$ (i.e., have at least one endpoint in $\mathcal{A}_t$) are all exactly those in $E^*_{\mathcal{A}_t} \cup E_{\mathcal{A}_t \cap \mathcal{D}_p}$. It follows that each J_t is an induced subgraph of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$, and these induced subgraphs are disjoint and cover all vertices of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$. Therefore, in what follows, we do not distinguish between the vertices of each J_t and their corresponding vertices in $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$. Our next plan is to construct a tree decomposition for $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$ of width $O(p)$ by properly gluing tree decompositions of the induced subgraphs J_t. To this end, we first observe that $\mathbf{tw}(J_t) = O(p)$ for all $t \in T_{\mathsf{supp}}$. Consider a node $t \in T_{\mathsf{supp}}$ at the jth level. Since $\mathcal{A}_t \subseteq \mathcal{L}_{[i_j+1, i_{j+1}]}$, $G_{\mathcal{A}_t}/E^*_{\mathcal{A}_t}$ is a subgraph of $G_{\mathcal{L}_{[i_j+1,i_{j+1}]}}/E^*_{\mathcal{L}_{[i_j+1,i_{j+1}]}}$. By property (iii) of Lemma 3.2, we have the inequality

$$\mathbf{tw}(J_t) \leq \mathbf{tw}(G_{\mathcal{A}_t}/E^*_{\mathcal{A}_t}) \leq \mathbf{tw}\left(G_{\mathcal{L}_{[i_j+1,i_{j+1}]}} \middle/ E^*_{\mathcal{L}_{[i_j+1,i_{j+1}]}}\right) = O(p).$$

Therefore, for each $t \in T_{\mathsf{supp}}$, there exists a tree decomposition (T^*_t, β^*_t) for J_t of width $O(p)$.[3] By Observation 3.10, $G_{\mathcal{A}_t}$ and $G_{\mathcal{A}_{t'}}$ are adjacent in $G_{\mathcal{D}}$ (i.e., there exists an edge of $G_{\mathcal{D}}$ with one endpoint in $G_{\mathcal{A}_t}$ and the other endpoint in $G_{\mathcal{A}_{t'}}$) only if t and t' are adjacent nodes in T_{supp}. It follows that two induced subgraphs J_t and $J_{t'}$ of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$ are adjacent only if t and t' are adjacent nodes in T_{supp}. Furthermore, we notice the following fact.

Observation 3.11. *For two nodes $t, s \in T_{\mathsf{supp}}$ where t is the parent of s, there exists at most one vertex in J_t that is neighboring to J_s in $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$.*

Proof. Suppose s is at the jth level of T_{supp}, and thus t is at the $(j-1)$-th level. By construction, all unit disks in $\mathcal{A}_s$ lie in the same connected component of $G_{\mathcal{L}_{>i_j}}$. Thus, by property (ii) of Lemma 3.2, the unit disks in $\mathcal{L}_{i_j}$ that are neighboring to $G_{\mathcal{A}_s}$ lie in the same connected component of $G_{\mathcal{L}_{i_j}}$, and hence the same connected component of $G_{\mathcal{D}_p}$. Note that all vertices of $G_{\mathcal{A}_t}$ that are neighboring to $G_{\mathcal{A}_s}$ must lie in $\mathcal{L}_{i_j}$, by property (i) of Lemma 3.2. Therefore, the vertices in $G_{\mathcal{A}_t}$ neighboring

[3] If t is the root of T_{supp}, then J_t is an empty graph. In this case, we simply let T^*_t be the tree with a single node x and set $\beta^*_t(x) = \emptyset$.

to $G_{\mathcal{A}_s}$ (if any) are contained in the same connected component of $G_{\mathcal{D}_p}$. By Observation 3.9, the vertices of this connected component are all contained in $\mathcal{A}_t$, and thus are contracted into one vertex in J_t, which is the only vertex in J_t that can be neighboring to J_s in $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$. □

Based on the above observation, we glue the tree decompositions (T^*_t, β^*_t) along the edges of T_{supp} to obtain a tree decomposition (T^*, β^*) for $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$ as follows. Consider a non-root node $s \in T_{\mathsf{supp}}$ with parent t. By Observation 3.11, there is at most one vertex v of J_t that is neighboring to J_s in $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$. We pick a node $t^* \in T^*_t$ whose bag $\beta^*_t(t^*)$ contains v, and call t^* the *portal* of s. (If no vertex of J_t is neighboring to J_s, we simply pick an arbitrary node $t^* \in T^*_t$ as the portal of s.) We then add an edge to connect the root of T^*_s and the portal t^*. We do this for all non-root nodes of T_{supp}. After that, we glue all trees in $\{T^*_t\}_{t \in T_{\mathsf{supp}}}$ together and obtain the new tree T^*. Next, we associate to each node $s^* \in T^*$ a bag $\beta^*(s^*)$ as follows. Consider a node $s^* \in T^*$ and suppose s^* originally belongs to T^*_s for $s \in T_{\mathsf{supp}}$. If s is the root, we simply define $\beta^*(s^*) = \beta^*_s(s^*) = \emptyset$. If s is not the root, let t be the parent of s in T_{supp} and $t^* \in T^*_t$ be the portal of s. We then define $\beta^*(s^*) = \beta^*_s(s^*) \cup \beta^*_t(t^*)$.

Observation 3.12. *(T^*, β^*) is a tree decomposition of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$ of width $O(p)$.*

Proof. Since the widths of the tree decompositions (T^*_t, β^*_t) are all $O(p)$, the size of each bag of (T^*, β^*) is bounded by $O(p)$ by our construction. So it suffices to show that (T^*, β^*) is a tree decomposition of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$. First, every vertex v of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$ is contained in some bag of (T^*_t, β^*_t), for some $t \in T_{\mathsf{supp}}$. Indeed, v belongs to J_t for some $t \in T_{\mathsf{supp}}$ and hence there exists a node $t^* \in T^*_t$ such that $v \in \beta^*_t(t^*)$, because (T^*_t, β^*_t) is a tree decomposition of J_t. By our construction, t^* is also a node of T^*, and we have $v \in \beta^*(t^*)$. Second, we show that for every edge (u, v) of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$, there exists some node of T^* whose bag contains both u and v. If (u, v) is an edge in some J_t, then there exists $t^* \in T^*_t$ such that $u, v \in \beta^*_t(t^*)$, as (T^*_t, β^*_t) is a tree decomposition of J_t. In this case, we have $u, v \in \beta^*(t^*)$. The other case is that (u, v) is an edge between two induced subgraphs J_s and J_t of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$. As we noticed before Observation 3.11, in this case, s and t are adjacent nodes in T_{supp}. Without loss of generality, assume t is the parent of s in T_{supp} and u (resp., v) lies in J_s (resp., J_t). By Observation 3.11, v is the only vertex in J_t that is neighboring to J_s. Let $t^* \in T^*_t$ be the portal of s. According to our choice of the portals, we have $v \in \beta^*_t(t^*)$. Now pick any node $s^* \in T^*_s$ such that $u \in \beta^*_s(s^*)$. By construction, we have $\beta^*(s^*) = \beta^*_s(s^*) \cap \beta^*_t(t^*)$ and hence $u, v \in \beta^*(s^*)$. Finally, we show that for any vertex v of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$, the nodes of T^* whose bag contains v are connected in T^*. Suppose v is in J_t for some $t \in T_{\mathsf{supp}}$. Observe that v is contained in the bags of two types of nodes in T^*. The first type are the nodes which originally belong to T^*_t and whose bags in $\mathcal{T}^*_t$ contain v; this type of nodes are connected in T^* as they are connected in T^*_t. The second type are all nodes which originally belong to T^*_s for some child s of t such that the bag of the portal of s contains v. Note that T^*_s is connected in T^* and the portal of s is a node of the first type as its bag contains v. Therefore, the nodes of the second type form some connected parts in T^* each of which is adjacent to a node of the first type. It follows that the nodes of T^* whose bags contain v are connected in T^*. As a result, (T^*, β^*) is a tree decomposition of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_p})$. □

3.4 Handling the General Case

In the previous section, we have proved that the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ satisfies the condition in Theorem 3.1 for the special case where $\mathcal{D}' = \emptyset$. In this section, we shall consider the general case and complete the proof for Theorem 3.1. Let $i \in [p]$. Our goal is to show $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})) = O(p + |\mathcal{D}'|)$ for every $\mathcal{D}' \subseteq \mathcal{D}_i$, knowing $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})) = O(p)$.

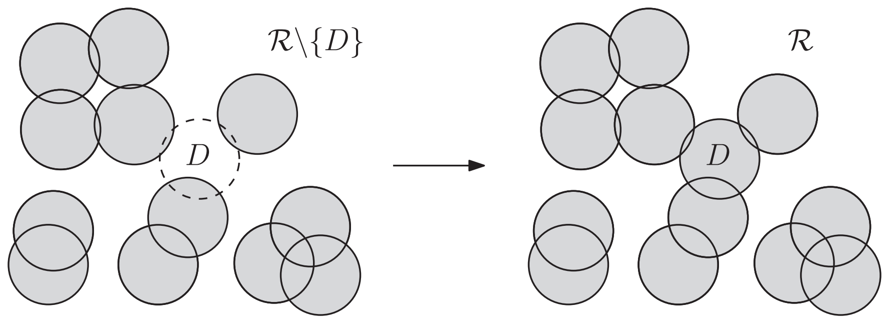

Fig. 4. The three components of $G_{\mathcal{R}\backslash\{D\}}$ hit by D are merged into one connected component in $G_{\mathcal{R}}$, while the others remain the same.

For convenience, we denote by V the vertex set of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ and V' the vertex set of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i\backslash\mathcal{D}'})$. Since $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ is obtained from $G_{\mathcal{D}}$ via edge contraction, there is a corresponding quotient map $\pi : \mathcal{D} \to V$. Similarly, there is a quotient map $\pi' : \mathcal{D} \to V'$ corresponding to the edge contraction for obtaining $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i\backslash\mathcal{D}'})$. Note that $E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i\backslash\mathcal{D}'} \subseteq E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i}$. So there exists a unique map $\rho : V' \to V$ such that $\pi = \rho \circ \pi'$, and $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ can be viewed as a graph obtained from $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i\backslash\mathcal{D}'})$ via edge contraction with quotient map ρ.

As $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})) = O(p)$, there exists a tree decomposition (T, β) of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ of width $O(p)$. We define a map $\beta' : T \to 2^{V'}$ as $\beta'(t) = \rho^{-1}(\beta(t))$ for all nodes $t \in T$. By Fact 2.2, (T, β') is a tree decomposition of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i\backslash\mathcal{D}'})$. Now it suffices to show that the width of this tree decomposition is $O(p + |\mathcal{D}'|)$. To this end, we establish a basic property of unit-disk graphs. For a graph G, we use the notation $\|G\|$ to denote the number of connected components of G. We have the following lemma.

Lemma 3.13. *For a set $\mathcal{R}$ of unit disks and $\mathcal{R}' \subseteq \mathcal{R}$, $\|G_{\mathcal{R}\backslash\mathcal{R}'}\| - \|G_{\mathcal{R}}\| = O(|\mathcal{R}'|)$.*

Proof. We show that $\|G_{\mathcal{R}\backslash\{D\}}\| - \|G_{\mathcal{R}}\| = O(1)$ for any unit disk $D \in \mathcal{R}$. Then the lemma can be proved via a simple induction argument. We say D *hits* a connected component of $G_{\mathcal{R}\backslash\{D\}}$ if D intersects some unit disk in this connected component. Note that all connected components of $G_{\mathcal{R}\backslash\{D\}}$ hit by D are merged into one connected component in $G_{\mathcal{R}}$, and all the other connected components of $G_{\mathcal{R}\backslash\{D\}}$ remain the same in $G_{\mathcal{R}}$. See Figure 4 for an example. Thus, the quantity $\|G_{\mathcal{R}\backslash\{D\}}\| - \|G_{\mathcal{R}}\|$ is equal to the number of connected components of $G_{\mathcal{R}\backslash\{D\}}$ hit by D minus 1. So it suffices to show that D only hits $O(1)$ connected components of $G_{\mathcal{R}\backslash\{D\}}$. Suppose D hits k connected components of $G_{\mathcal{R}\backslash\{D\}}$. Pick a unit disk from each such connected component, and let $D_1, \ldots, D_k$ be these unit disks. Note that $D_1, \ldots, D_k$ are disjoint as they are from different connected components of $G_{\mathcal{R}\backslash\{D\}}$. Thus, $D, D_1, \ldots, D_k$ form an induced biclique $K_{1,k}$ in $G_{\mathcal{R}}$. It was known that unit-disk graphs exclude $K_{1,6}$ as an induced subgraph [5]. So we have $k \leq 5 = O(1)$. □

Using the above lemma, we show that $|\rho^{-1}(U)| = O(|U| + |\mathcal{D}'|)$ for any $U \subseteq V$. Since $\mathcal{D}_i$ is a grid-respecting subset of $\mathcal{D}$, for each $v \in V$, $\pi^{-1}(\{v\})$ is either (the vertex set of) a cell clique of $G_{\mathcal{D}}$ that is disjoint from $\mathcal{D}_i$ or (the vertex set of) a connected component of $G_{\mathcal{D}_i}$; we say v is a *type-1* vertex in the former case and a *type-2* vertex in the latter case. Let U_1 (resp., U_2) be the type-1 (resp., type-2) vertices in U. For each $u \in U_1$, we have $|\rho^{-1}(\{u\})| = |\pi'(\pi^{-1}(\{u\}))| = 1$, as every cell clique of $G_{\mathcal{D}}$ is contracted into one vertex in $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i\backslash\mathcal{D}'})$. Thus, $|\rho^{-1}(U_1)| = |U_1|$. To bound $|\rho^{-1}(U_2)|$, we consider $\pi^{-1}(U_2) \subseteq \mathcal{D}$. By definition, $\pi^{-1}(\{u\})$ is a connected component of

$G_{\mathcal{D}_i}$ for each $u \in U_2$, and thus $\|G_{\pi^{-1}(U_2)}\| = |U_2|$. Set $\mathcal{I} = \pi^{-1}(U_2) \cap \mathcal{D}'$. By Lemma 3.13, we have

$$\|G_{\pi^{-1}(U_2)\backslash\mathcal{D}'}\| - \|G_{\pi^{-1}(U_2)}\| = \|G_{\pi^{-1}(U_2)\backslash\mathcal{I}}\| - \|G_{\pi^{-1}(U_2)}\| = O(|\mathcal{I}|),$$

which implies $\|G_{\pi^{-1}(U_2)\backslash\mathcal{D}'}\| = O(|U_2| + |\mathcal{D}'|)$ because $|\mathcal{I}| \leq |\mathcal{D}'|$. Since $\pi^{-1}(U_2)\backslash\mathcal{D}' \subseteq \mathcal{D}_i\backslash\mathcal{D}'$, π' maps the vertices in each connected component of $G_{\pi^{-1}(U_2)\backslash\mathcal{D}'}$ to the same vertex in V'. Therefore, $|\pi'(\pi^{-1}(U_2)\backslash\mathcal{D}')| \leq \|G_{\pi^{-1}(U_2)\backslash\mathcal{D}'}\| = O(|U_2| + |\mathcal{D}'|)$. Now we have the inequality

$$|\pi'(\pi^{-1}(U_2))| \leq |\pi'(\pi^{-1}(U_2)\backslash\mathcal{D}')| + |\pi'(\mathcal{D}')| = O(|U_2| + |\mathcal{D}'|).$$

It follows that $|\rho^{-1}(U_2)| = O(|U_2| + |\mathcal{D}'|)$, and thus $|\rho^{-1}(U)| = O(|U| + |\mathcal{D}'|)$. As a result, for all $t \in T$, $|\beta'(t)| = |\rho^{-1}(\beta(t))| = O(|\beta(t)| + |\mathcal{D}'|) = O(p + |D'|)$. So (T, β') is a tree decomposition of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i\backslash\mathcal{D}'})$ of width $O(p + |D'|)$, completing the proof of Theorem 3.1.

4 Applications

4.1 Contraction Decomposition for Unit-Disk Graphs

In this section, we use Theorem 3.1 to prove the first CDT for unit-disk graphs, which is shown below.

Theorem 4.1 (CDT). *Given a set $\mathcal{D}$ of n unit disks and an integer $p \in [n]$, one can compute in polynomial time a partition $\{E_1, \ldots, E_p\}$ of $E_{\mathcal{D}}$ such that for every $i \in [p]$, $\mathbf{tw}(G_{\mathcal{D}}/E_i) = O(p^2)$.*

Observe that to prove the above theorem, it suffices to compute in polynomial time p *disjoint* subsets $E_1, \ldots, E_p$ of edges of $G_{\mathcal{D}}$ such that $\mathbf{tw}(G_{\mathcal{D}}/E_i) = O(p^2)$ for every $i \in [p]$ (that is, we do not require $\{E_1, \ldots, E_p\}$ to be a partition of the edge set $E_{\mathcal{D}}$). Indeed, we can arbitrarily assign the remaining edges $E_{\mathcal{D}} \backslash \bigcup_{i=1}^{p} E_i$ to the p subsets to obtain a partition $\{E'_1, \ldots, E'_p\}$ such that $E_i \subseteq E'_i$ for all $i \in [p]$, and then $\mathbf{tw}(G_{\mathcal{D}}/E'_i) \leq \mathbf{tw}(G_{\mathcal{D}}/E_i) = O(p^2)$.

We start by applying the algorithm of Theorem 3.1 on $\mathcal{D}$ to obtain in polynomial time a grid-respecting partition $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$ of $\mathcal{D}$. Consider any $i \in [p]$. Setting $\mathcal{D}' = \emptyset$ in Theorem 3.1 gives us $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})) = O(p)$. We are going to use this fact later in our analysis. Next, we state a lemma which will be used in our construction of the edge sets $E_1, \ldots, E_p$.

Lemma 4.2. *The edge set of a clique K of size larger than $4p$ can be partitioned in polynomial time into p parts such that each part contains a spanning tree of K.*

Proof. It is well-known [46] that there exists a polynomial-time algorithm to compute a Hamiltonian path in a graph G where the degree of each vertex is at least half of the total number of vertices. We shall use this algorithm to prove the lemma and call it *Palmer's algorithm* for ease of reference.

Let $q > 4p$ be the size of K. We first compute p edge-disjoint Hamiltonian paths in K over p iterations. Set $K_1 = K$. In iteration $i \in [p]$, we compute a Hamiltonian path H_i in K_i by applying Palmer's algorithm on K_i. We then remove the edges of H_i from K_i to obtain the graph K_{i+1}. To see that the condition for applying Palmer's algorithm on K_i is always satisfied for every $i \in [p]$, note that through the first $i-1$ iterations, the degree of a vertex decreases by at most $2(i-1)$. Indeed, in each iteration we remove a Hamiltonian path from the current graph and thus the degree of a vertex decreases by at most 2. Thus, in iteration i, the degree of any vertex is at least $(q-1) - 2(i-1) = q + 1 - 2i \geq q + 1 - 2p > q/2$. The last inequality follows as $q > 4p$. Finally, we add all the edges of K_{p+1} to H_p so that $\{H_1, \ldots, H_p\}$ form a partition of the edges of K and each H_i contains a spanning tree of K as it contains a Hamiltonian path in K. Our algorithm runs in polynomial time as Palmer's algorithm does. □

We construct the edge sets $E_1, \dots, E_p$ in the following way. Consider any edge $e = (u, v) \in E_{\mathcal{D}}$. If $u \in \mathcal{D}_i$ and $v \in \mathcal{D}_j$ for $i \neq j$, then we totally ignore e (i.e., do not add it to any of $E_1, \dots, E_p$). Otherwise, let $u, v \in \mathcal{D}_i$ for some $i \in [p]$. If e is not a part of any cell clique, we add e to the part E_i. If e is a part of a cell clique of size at most $4p$, we also add e to the part E_i. The only remaining edges are those in the cell cliques of size larger than $4p$. Consider any such cell clique K. Using the algorithm in Lemma 4.2, we partition the edge set of K into exactly p parts $H_1, \dots, H_p$ each of which contains a spanning tree of K, and then add the edges in H_i to E_i for $i \in [p]$. This completes the construction of $E_1, \dots, E_p \subseteq E_{\mathcal{D}}$. It is clear that $E_1, \dots, E_p$ are disjoint. Now it suffices to bound $\mathbf{tw}(G_{\mathcal{D}}/E_i)$ for every $i \in [p]$.

Lemma 4.3. *For all $i \in [p]$, $\mathbf{tw}(G_{\mathcal{D}}/E_i) = O(p^2)$.*

Proof. Let V^* be the vertex set of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ and V be the vertex set of $G_{\mathcal{D}}/E_i$. By our construction, we have $E_i \subseteq E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i}$. Thus, $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ can be obtained from $G_{\mathcal{D}}/E_i$ via edge contraction, and let $\rho : V \to V^*$ be the corresponding quotient map. Recall that $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})) = O(p)$. So there exists a tree decomposition (T^*, β^*) of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ of width $O(p)$. By Fact 2.2, (T^*, β) is a tree decomposition of $G_{\mathcal{D}}/E_i$ where $\beta : T^* \to 2^V$ is defined as $\beta(t^*) = \rho^{-1}(\beta^*(t^*))$ for $t^* \in T^*$. We show that $|\rho^{-1}(\{v^*\})| = O(p)$ for every $v^* \in V^*$, which implies

$$|\beta(t^*)| = |\rho^{-1}(\beta^*(t^*))| = O(p \cdot |\beta^*(t^*)|) = O(p^2),$$

and hence completes the proof.

Let $\pi_1 : \mathcal{D} \to V^*$ (resp., $\pi_2 : \mathcal{D} \to V$) be the quotient map corresponding to the edge contraction for obtaining $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i})$ (resp., $G_{\mathcal{D}}/E_i$). We have $\pi_1 = \rho \circ \pi_2$. Consider a vertex $v^* \in V^*$. Since $\mathcal{D}_i$ is grid-respecting, for each vertex $v^* \in V^*$, $\pi_1^{-1}(\{v^*\})$ is either (the vertex set of) a cell clique in $G_{\mathcal{D}}$ that is disjoint from $\mathcal{D}_i$ or (the vertex set) a connected component of $G_{\mathcal{D}_i}$. In the former case, if $|\pi_1^{-1}(\{v^*\})| \leq 4p$, then $|\rho^{-1}(\{v^*\})| = |\pi_2(\pi_1^{-1}(\{v^*\}))| \leq |\pi_1^{-1}(\{v^*\})| \leq 4p$. If $|\pi_1^{-1}(\{v^*\})| > 4p$, then we have $|\rho^{-1}(\{v^*\})| = |\pi_2(\pi_1^{-1}(\{v^*\}))| = 1$, since E_i contains a spanning tree of any cell clique in $G_{\mathcal{D}}$ of size larger than $4p$. In the latter case, $\pi_1^{-1}(\{v^*\})$ is a connected component of $G_{\mathcal{D}_i}$, and we claim $|\rho^{-1}(\{v^*\})| = 1$. Indeed, E_i contains all edges in a connected component of $G_{\mathcal{D}_i}$ except some edges in cell cliques of size larger than $4p$. But for each of these large cell cliques, E_i contains at least one of its spanning trees (which connects all vertices in the clique). Therefore, a connected component of $G_{\mathcal{D}_i}$ is contracted into a single vertex in $G_{\mathcal{D}}/E_i$, which implies $|\rho^{-1}(\{v^*\})| = 1$. It follows that $|\rho^{-1}(\{v^*\})| = O(p)$ for every $v^* \in V^*$. □

4.2 Near-Optimal Bipartization for Unit-Disk Graphs

In the Bipartization problem, we are given a graph $G = (V, E)$ as well as a parameter k, and the goal is to decide whether there exists a subset $X \subseteq V$ of size at most k such that $G - X$ is bipartite. We will sometime use the term *left part* or *right part* to denote the two parts of the bipartite graph $G - X$. This problem is sometimes referred to as OCT, and X is called an *OCT* of G as bipartite graphs are exactly graphs without odd cycles. Equivalently, we can also formulate Bipartization in the following way. Consider a map $\lambda : V \to \{0, 1, 2\}$. We say λ is *bipartite* if for every edge $(u, v) \in E$, neither $\lambda(u) = \lambda(v) = 1$ nor $\lambda(u) = \lambda(v) = 2$. The cost of λ, denoted by $\text{cost}(\lambda)$ is, the number of vertices in V that are mapped to 0, i.e., $\text{cost}(\lambda) = |\lambda^{-1}(\{0\})|$. Then the Bipartization problem is equivalent to finding a bipartite map $\lambda : V \to \{0, 1, 2\}$ of cost at most k. Indeed, the OCT $X \subseteq V$ is nothing but $\lambda^{-1}(\{0\})$, while $\lambda^{-1}(\{1\})$ and $\lambda^{-1}(\{2\})$ are the left and right parts of the graph $G - X$, respectively.

The map λ defined above can be directly generalized to any subset of V. Formally, a *configuration* on a subset $V' \subseteq V$ is a map $\lambda : V' \to \{0, 1, 2\}$. A configuration is *bipartite* if for every $(u, v) \in E$ with $u, v \in V'$, neither $\lambda(u) = \lambda(v) = 1$ nor $\lambda(u) = \lambda(v) = 2$. Two configurations $\lambda_1 : V_1 \to \{0, 1, 2\}$ and

$\lambda_2 : V_2 \rightarrow \{0,1,2\}$ are *compatible* if $\lambda_1(v) = \lambda_2(v)$ for all $v \in V_1 \cap V_2$. If $\lambda_1, \ldots, \lambda_m$ are configurations on $V_1, \ldots, V_m \subseteq V$ that are pairwise compatible, one can "glue" them to obtain a configuration $\lambda : \bigcup_{i=1}^{m} V_i \rightarrow \{0,1,2\}$ satisfying $\lambda_{|V_i} = \lambda_i$ for all $i \in [k]$. With these notions defined, let us consider the problem in hand. Let $\mathcal{D}$ be a set of n unit disks, and we want to solve Bipartization on $G_{\mathcal{D}}$.

An easy but crucial remark is that, for every clique K in $G_{\mathcal{D}}$, the OCT contains all vertices of K except at most two. The algorithm starts by checking if there is some cell clique with size at least $k+3$, in which case it trivially answers NO. From now on, we may assume all cell cliques have size at most $k+2$. The first step of our algorithm is to apply the following randomized algorithm to obtain a small candidate set $\mathsf{Cand} \subseteq \mathcal{D}$ for OCT. This can be done via the technique of representative sets, see Lemma 5 in [6] for more details.

Lemma 4.4. *Given a graph $G = (V, E)$ and a number k, one can compute $\mathsf{Cand} \subseteq V$ of size $k^{O(1)}$ such that G has an OCT of size k if and only if G has an OCT of size k that is a subset of Cand, using a polynomial-time randomized algorithm with success probability $1 - 1/2^{|V|}$.*

By the above lemma, $|\mathsf{Cand}| = k^{O(1)}$ and it suffices to find an OCT of $G_{\mathcal{D}}$ in Cand of size at most k. Suppose there exists an (unknown) OCT $\mathcal{X} \subseteq \mathsf{Cand}$ of size at most k. Next, we apply the algorithm of Theorem 3.1 with $p = \lfloor \sqrt{k} \rfloor$ to obtain the grid-respecting partition $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$ of $\mathcal{D}$ in polynomial time. As the OCT $\mathcal{X}$ we are looking for is of size at most k and $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$ is a partition of $\mathcal{D}$, there exists an index $i \in [p]$ such that $|\mathcal{D}_i \cap \mathcal{X}| \leq k/p$. By trying all indices in $[p]$, we can assume that the algorithm knows the index i. Moreover, we know that $\mathcal{D}_i \cap \mathcal{X} \subseteq \mathcal{D}_i \cap \mathsf{Cand}$ as $\mathcal{X} \subseteq \mathsf{Cand}$. Thus, by trying all the subsets of $\mathcal{D}_i \cap \mathsf{Cand}$ of size at most k/p, we can assume that the algorithm knows $\mathcal{S} = \mathcal{D}_i \cap \mathcal{X}$; note that the number of such subsets is bounded by $|\mathsf{Cand}|^{O(k/p)} = 2^{O(\sqrt{k}\log k)}$.

Now it suffices to find an OCT of size at most k which intersects $\mathcal{D}_i$ at $\mathcal{S}$, i.e., contains $\mathcal{S}$ but is disjoint from $\mathcal{D}_i \backslash \mathcal{S}$. We now use the language of configuration to formulate this task. We say a configuration $\lambda : \mathcal{D}' \rightarrow \{0,1,2\}$ on $\mathcal{D}' \subseteq \mathcal{D}$ is *valid* if it is bipartite and $\lambda(D) = 0$ (resp., $\lambda(D) \neq 0$) for all $D \in \mathcal{D}' \cap \mathcal{S}$ (resp., $D \in \mathcal{D}' \cap (\mathcal{D}_i \backslash \mathcal{S})$). Then our goal is nothing but finding a valid configuration on $\mathcal{D}$ of cost at most k. Alternatively, our algorithm finds a *minimum-cost* valid configuration $\lambda^* : \mathcal{D} \rightarrow \{0,1,2\}$.

The idea for efficiently finding λ^* is to use the fact $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})) = O(p + |\mathcal{S}|) = O(\sqrt{k})$, which follows from Theorem 3.1, together with a widely used technique—DP on tree decomposition. To apply this idea, however, there is one difficulty to be overcome: we need to do DP on a tree decomposition of $G_{\mathcal{D}}$, but the graph of $O(\sqrt{k})$ treewidth is $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})$ instead of $G_{\mathcal{D}}$.

Let V^* be the vertex set of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})$ and $\pi : \mathcal{D} \rightarrow V^*$ be the quotient map of the edge contraction for obtaining $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})$. Since $\mathbf{tw}(G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})) = O(\sqrt{k})$, by Lemma 2.1, we can compute in $2^{O(\sqrt{k})} n^{O(1)}$ time a binary tree decomposition (T, β^*) of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})$ of width $O(\sqrt{k})$. By Fact 2.2, (T, β) is a binary tree decomposition of $G_{\mathcal{D}}$, where $\beta(t) = \pi^{-1}(\beta^*(t))$ for $t \in T$. We shall do DP on (T, β). Note that the bags $\beta(t)$ of (T, β) can be large. The key trick here is to argue that the number of valid configurations on each bag $\beta(t)$ is small, specifically bounded by $2^{O(\sqrt{k}\log k)}$, by using the $O(\sqrt{k})$ width of (T, β^*). We first observe the following simple fact.

Observation 4.5. *For each vertex $v^* \in V^*$, $\pi^{-1}(\{v^*\})$ is either a cell clique of $G_{\mathcal{D}}$ or a connected component of $G_{\mathcal{D}_i \backslash \mathcal{S}}$ together with some elements in $\mathcal{S}$.*

Proof. Let $v^* \in V^*$ and $D \in \mathcal{D}$ such that $\pi(D) = v^*$. If $(\mathcal{D}_i \backslash \mathcal{S}) \Cap \square_D = \emptyset$, then the only edges in $E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}}$ incident to the vertices in the cell clique of $\square_D$ are those in the cell clique. In this case, $\pi^{-1}(\{v^*\})$ is the cell clique of $\square_D$. The remaining case is that $(\mathcal{D}_i \backslash \mathcal{S}) \Cap \square_D$ contains at least one unit disk D'. Consider the connected component of $G_{\mathcal{D}_i \backslash \mathcal{S}}$ containing D' and let $C \subseteq \mathcal{D}_i \backslash \mathcal{S}$ be the vertex set of this connected component. Define $C^+ = \bigcup_{C \in C} (\mathcal{D} \Cap \Box_C)$. We claim that $\pi^{-1}(v^*) = C^+$. First, it is clear that $C^+ \subseteq \pi^{-1}(v^*)$, because the unit disks in C^+ are connected by the edges in $E^*_{\mathcal{D}} \cup E_C \subseteq E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}}$. On the other hand, since C forms a connected component of $G_{\mathcal{D}_i \backslash \mathcal{S}}$, all edges in $E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}}$ incident to C^+ are actually in E_{C^+}. Thus, $\pi^{-1}(v^*) = C^+$. Finally, note that $C^+ \backslash C \subseteq \mathcal{S}$, because $\mathcal{D}_i$ is grid-respecting (hence $C^+ \subseteq \mathcal{D}_i$) and a unit disk in $C^+ \backslash C$ cannot be in $\mathcal{D}_i \backslash \mathcal{S}$ (as C forms a connected component of $G_{\mathcal{D}_i \backslash \mathcal{S}}$). Therefore, in this case, $\pi^{-1}(\{v^*\})$ is a connected component of $G_{\mathcal{D}_i \backslash \mathcal{S}}$ together with some elements in $\mathcal{S}$. □

Lemma 4.6. *For every $t \in T$, the number of valid configurations on $\beta(t)$ is $2^{O(\sqrt{k} \log k)}$. Furthermore, these valid configurations can be constructed in $2^{O(\sqrt{k} \log k)} n^{O(1)}$ time.*

Proof. Observe that if $\lambda : \beta(t) \to \{0, 1, 2\}$ is a valid configuration on $\beta(t)$, then for any subset $\mathcal{A} \subseteq \beta(t)$, $\lambda_{|\mathcal{A}}$ is a valid configuration on $\mathcal{A}$. Therefore, if we use $\xi(\mathcal{A})$ to denote the number of valid configurations on a subset $\mathcal{A} \subseteq \beta(t)$, then we have $\xi(\beta(t)) \leq \prod_{v^* \in \beta^*(t)} \xi(\pi^{-1}(\{v^*\}))$, because $\{\pi^{-1}(\{v^*\}) : v^* \in \beta^*(t)\}$ is a partition of $\beta(t) = \pi^{-1}(\beta^*(t))$. By Observation 4.5, for every $v^* \in \beta^*(t)$, $\pi^{-1}(\{v^*\})$ is either a cell clique of $G_{\mathcal{D}}$ or a connected component of $G_{\mathcal{D}_i \backslash \mathcal{S}}$ together with some elements in $\mathcal{S}$. In the former case, recall our assumption that all cell cliques have size at most $k + 2$. Also, a valid configuration on a cell clique must map all but at most two vertices to 0. Thus, $\xi(\pi^{-1}(\{v^*\})) = O(k^2)$ in this case. In the latter case, observe that a valid configuration must map all vertices in the connected component of $G_{\mathcal{D}_i \backslash \mathcal{S}}$ to $\{1, 2\}$ and map all vertices in $\pi^{-1}(\{v^*\}) \cap \mathcal{S}$ to 0. Furthermore, there are only two ways a valid configuration can map the vertices in the connected component to $\{1, 2\}$, under the bipartite restriction, i.e., two adjacent vertices cannot be both mapped to 1 or 2 (once we fix the label of one vertex in the connected component, the labels of the other vertices are uniquely determined). Thus, $\xi(\pi^{-1}(\{v^*\})) = 2$ in this case. Finally, as $|\beta^*(t)| = O(\sqrt{k})$, we have $\xi(\beta(t)) \leq \prod_{v^* \in \beta^*(t)} \xi(\pi^{-1}(\{v^*\})) = k^{O(\sqrt{k})} = 2^{O(\sqrt{k} \log k)}$. To construct the valid configurations on $\beta(t)$, we can construct the valid configurations on each $\pi^{-1}(\{v^*\})$ and then glue them, which can be done in $2^{O(\sqrt{k} \log k)} n^{O(1)}$ time. □

With Lemma 4.6 in hand, the remaining part of our algorithm just follows the standard DP on (T, β). For each $t \in T$, let $\gamma(t)$ be the union of the bags of all nodes in the subtree T_t of T rooted at t. We compute a DP table at t, in which each entry corresponds to a valid configuration $\lambda_t : \beta(t) \to \{0, 1, 2\}$ on $\beta(t)$. The entry corresponding to λ_t stores a minimum-cost valid configuration on $\gamma(t)$ that is compatible with λ_t. By Lemma 4.6, the size of the DP table is $2^{O(\sqrt{k} \log k)}$ and the valid configurations on $\beta(t)$ corresponding to the table entries can be computed in $2^{O(\sqrt{k} \log k)} n^{O(1)}$ time. As usual, we fill out the DP tables at the nodes in T in a bottom-up fashion. The tables at the leaves of T can be filled out in a trivial way, since $\gamma(t) = \beta(t)$ for a leaf $t \in T$. Consider a non-leaf node $t \in T$ with left child l and right child r. Suppose we already have the DP tables at l and r, and we are going to fill out the DP table at t. Specifically, for each valid configuration $\lambda_t : \beta(t) \to \{0, 1, 2\}$, we want to find a minimum-cost valid configuration $\lambda'_t : \gamma(t) \to \{0, 1, 2\}$ compatible with λ_t.

Observation 4.7. *Let $B = \beta(t) \cup \beta(l) \cup \beta(r)$ and $\lambda : B \to \{0, 1, 2\}$ be a valid configuration on B. If $\lambda'_l : \gamma(l) \to \{0, 1, 2\}$ and $\lambda'_r : \gamma(r) \to \{0, 1, 2\}$ are valid configurations both compatible with λ, then λ'_l and λ'_r are compatible, and furthermore the configuration $\lambda'_t : \gamma(t) \to \{0, 1, 2\}$ obtained by gluing $\lambda, \lambda'_l, \lambda'_r$ is valid and satisfies*

$$\mathsf{cost}(\lambda'_t) = \mathsf{cost}(\lambda'_l) + \mathsf{cost}(\lambda'_r) + \Delta_\lambda,$$

where $\Delta_\lambda = \mathsf{cost}(\lambda_{|\beta(t)}) - \mathsf{cost}(\lambda_{|\beta(t) \cap \beta(l)}) - \mathsf{cost}(\lambda_{|\beta(t) \cap \beta(r)})$. In particular, if $\lambda'_l : \gamma(l) \to \{0, 1, 2\}$ (resp., $\lambda'_r : \gamma(r) \to \{0, 1, 2\}$) is a minimum-cost valid configuration of $\gamma(l)$ (resp., $\gamma(r)$) that is *compatible with λ, then the configuration $\lambda'_t : \gamma(t) \to \{0,1,2\}$ obtained by gluing $\lambda, \lambda'_l, \lambda'_r$ is a minimum-cost valid configuration of $\gamma(t)$ that is compatible with λ.*

Proof. To see λ'_l and λ'_r are compatible, we notice that $\gamma(l) \cap \gamma(r) \subseteq \beta(t)$. Indeed, if a unit disk $D \in \mathcal{D}$ is contained in both $\gamma(l)$ and $\gamma(r)$, then we must have $D \in \beta(t)$ as the nodes in T whose bag contains D are connected. As such, $\gamma(l) \cap \gamma(r) \subseteq B$. Since λ'_l and λ'_r are both compatible with λ, for any $D \in \gamma(l) \cap \gamma(r)$, we have $\lambda'_l(D) = \lambda(D) = \lambda'_r(D)$. Thus, λ'_l and λ'_r are compatible. Let $\lambda'_t : \gamma(t) \to \{0,1,2\}$ be obtained by gluing $\lambda, \lambda'_l, \lambda'_r$. Clearly, $\lambda'_t(D) = 0$ for any $D \in \gamma(t) \cap \mathcal{S}$ and $\lambda'_t(D) \neq 0$ for any $D \in \gamma(t) \cap (\mathcal{D}_i \backslash \mathcal{S})$, because all of $\lambda, \lambda'_l, \lambda'_r$ are valid. So it suffices to show that λ'_t is bipartite. Consider an edge $(D, D') \in E_{\mathcal{D}}$ with $D, D' \in \gamma(t)$. If $D, D' \in \beta(t)$, then we cannot have $\lambda'_t(D) = \lambda'_t(D') \in \{1,2\}$ as λ is bipartite. Otherwise, assume $D \notin \beta(t)$ without loss of generality. Then either $D \in \gamma(l)$ or $D \in \gamma(r)$; assume $D \in \gamma(l)$ without loss of generality. As the nodes in T whose bags containing D are connected, we know that D is only contained in the bags of the nodes in the subtree T_l rooted at l. Since $(D, D') \in E_{\mathcal{D}}$, there exists a node $s \in T$ such that $D, D' \in \beta(s)$. We must have $s \in T_l$ for $D \in \beta(s)$. This implies $D, D' \in \gamma(l)$. Because λ'_l is bipartite, we cannot have $\lambda'_t(D) = \lambda'_t(D') \in \{1,2\}$. Therefore, λ'_t is bipartite. The formula for $\text{cost}(\lambda'_t)$ follows easily from inclusion–exclusion principle.

Now suppose that $\lambda'_l : \gamma(l) \to \{0,1,2\}$ (resp., $\lambda'_r : \gamma(r) \to \{0,1,2\}$) is a minimum-cost valid configuration of $\gamma(l)$ (resp., $\gamma(r)$) that is compatible with λ. Let $\lambda'_t : \gamma(t) \to \{0,1,2\}$ obtained by gluing $\lambda, \lambda'_l, \lambda'_r$ be the configuration obtained by gluing $\lambda, \lambda'_l, \lambda'_r$. As argued above, λ'_t is valid and compatible with λ. Consider another valid configuration $\lambda''_t : \gamma(t) \to \{0,1,2\}$ that is compatible with λ. Let $\lambda''_l = {\lambda''_t}_{|\gamma(l)}$ and $\lambda''_l = {\lambda''_t}_{|\gamma(r)}$. Clearly, λ''_l and λ''_r are valid configurations of $\gamma(l)$ and $\gamma(r)$, and both of them are compatible with λ. Thus, $\text{cost}(\lambda''_l) \geq \text{cost}(\lambda'_l)$ and $\text{cost}(\lambda''_r) \geq \text{cost}(\lambda'_r)$. □

By the above observation, to find a minimum-cost valid configuration $\lambda'_t : \gamma(t) \to \{0,1,2\}$ compatible with a given valid configuration $\lambda : B \to \{0,1,2\}$ on $B = \beta(t) \cup \beta(l) \cup \beta(r)$, it suffices to find a minimum-cost valid configuration $\lambda'_l : \gamma(l) \to \{0,1,2\}$ (resp., $\lambda'_r : \gamma(r) \to \{0,1,2\}$) on $\gamma(l)$ (resp., $\gamma(r)$) that is compatible with λ and then glue $\lambda, \lambda'_l, \lambda'_r$. Note that $\gamma(l) \cap B = \beta(l)$ and $\gamma(r) \cap B = \beta(r)$, since the nodes in T whose bags contain a unit disk in $\mathcal{D}$ must be connected. Therefore, a minimum-cost valid configuration on $\gamma(l)$ (resp., $\gamma(r)$) compatible with λ is nothing but a minimum-cost valid configuration on $\gamma(l)$ (resp., $\gamma(r)$) compatible with $\lambda_{|\beta(l)}$ (resp., $\lambda_{|\beta(r)}$), which is stored in the DP table at l (resp., r). As such, a minimum-cost valid configuration on $\gamma(t)$ compatible with λ can be directly computed.

Recall that to fill out the DP table at t, what we want is a minimum-cost valid configuration on $\gamma(t)$ compatible with a valid configuration on $\beta(t)$ instead of B. However, we have $\beta(t) \subseteq B \subseteq \gamma(t)$. So a valid configuration $\lambda'_t : \gamma(t) \to \{0,1,2\}$ is compatible with a valid configuration $\lambda_t : \beta(t) \to \{0,1,2\}$ if and only if there exists a valid configuration $\lambda : B \to \{0,1,2\}$ compatible with both λ_t and λ'_t. Thus, given λ_t, we can construct all valid configurations $\lambda : B \to \{0,1,2\}$ compatible with λ_t, and for each λ compute a minimum-cost valid configuration on $\gamma(t)$ compatible with λ. By taking the minimum-cost one among the configurations on $\gamma(t)$ we compute, we obtain the desired minimum-cost valid configuration on $\gamma(t)$ compatible with λ_t. Note that the number of valid configurations on B is $2^{O(\sqrt{k}\log k)}$, as the numbers of valid configurations on $\beta(t), \beta(l), \beta(r)$ are all $2^{O(\sqrt{k}\log k)}$ by Lemma 4.6. Therefore, each entry of the DP table at t can be computed in $2^{O(\sqrt{k}\log k)}$ time. Then the entire DP table can be filled out in $2^{O(\sqrt{k}\log k)}$ time, as the size of the DP table is $2^{O(\sqrt{k}\log k)}$. In $2^{O(\sqrt{k}\log k)} n^{O(1)}$ time, we can finally complete the DP procedure. To find a minimum-cost valid configuration $\lambda^* : \mathcal{D} \to \{0,1,2\}$, we check the DP table at the root $\text{rt} \in T$. Note that $\gamma(\text{rt}) = \mathcal{D}$. Therefore, the minimum-cost one among the configurations stored in all entries of the DP table at rt is just the desired λ^*. This completes the discussion of our algorithm. The overall running time is $2^{O(\sqrt{k}\log k)} n^{O(1)}$, and the success probability is at least $1 - 1/2^{|\mathcal{D}|}$. So we conclude the following.

Theorem 4.8. *There exists a randomized algorithm that solves* Bipartization *on unit-disk graphs in $2^{O(\sqrt{k}\log k)} n^{O(1)}$ time, where n is the number of vertices and k is the solution size.*

We show that the algorithm in the above theorem is near optimal. Specifically, we cannot hope for a $2^{o(\sqrt{k})} n^{O(1)}$ running time, assuming ETH.

Theorem 4.9. *Assuming the ETH,* Bipartization *on unit-disk graphs cannot be solved in $2^{o(\sqrt{k})} n^{O(1)}$ time, where n is the number of vertices and k is the solution size.*

Proof. To show the desired lower bound, we give a reduction from Vertex Cover on unit-disk graphs to our problem. From the lower bound framework of de Berg et al. [13], it follows that Vertex Cover on unit-disk graphs cannot be solved in $2^{o(\sqrt{k})} n^{O(1)}$ time, unless ETH is false. Hence, it is sufficient to give a polynomial time parameter preserving reduction. Let $\mathcal{I}$ be any given instance of Vertex Cover on unit-disk graphs consisting of a set of n disks $\mathcal{D}$ in the plane, the corresponding unit-disk graph $G = (V, E)$, and a parameter k. First, we make a copy of all the disks in $\mathcal{D}$. Let us call this set $\mathcal{D}'$. Let G' be the unit-disk graph induced by the $2n$ disks in $\mathcal{D} \cup \mathcal{D}'$. As the disks in $\mathcal{D}$ are given to us, $G' = (V', E')$ can be constructed in polynomial time. We will prove that G has a vertex cover of size at most k if and only if G' has a solution to Bipartization of size at most $2k$.

First, suppose G has a vertex cover S of size $k_1 \leq k$. Note that $I = V \setminus S$ is an independent set of size $n - k_1$ in G. Thus, I corresponds to a subset $\mathcal{D}_1 \subseteq \mathcal{D}$ of disjoint disks in the plane. Let $\mathcal{D}_1'$ be the set of copies of the disks in $\mathcal{D}_1$. Thus, the unit-disk graph induced by $\mathcal{D}_1 \cup \mathcal{D}_1'$ is a matching of size $n - k_1$, and hence is an induced bipartite subgraph of G' having $2(n - k_1)$ vertices. Hence, G' has a solution to Bipartization of size $2k_1 \leq 2k$.

Now, suppose G' has a solution S' to Bipartization of size $k' \leq 2k$. Thus, the induced subgraph G'' of G' with $V' \setminus S'$ as the set of vertices, is bipartite. Note that $|V' \setminus S'| = 2n - k'$. Hence, G'' contains an independent set I'' of size at least $(2n - k')/2 \geq (2n - 2k)/2 \geq n - k$. However, I'' can be corresponding to a set $\mathcal{D}''$ of disks from both $\mathcal{D}$ and $\mathcal{D}'$. But the disks in $\mathcal{D}''$ are disjoint, and thus one can find another set of disjoint disks $\mathcal{D}_1'' \subseteq \mathcal{D}$ of size exactly $|I''|$ by replacing the disks of $\mathcal{D}'$ in $\mathcal{D}''$ by the corresponding original copies in $\mathcal{D}$. As the disks in $\mathcal{D}_1''$ are disjoint, G contains an independent set of size $|I''| \geq n - k$, and hence a vertex cover of size at most k. □

4.2.1 Generalization to GFVS with Non-Identity Labels. In fact, the previous algorithm can be generalized to the GFVS problem with non-identity labels on unit-disk graphs. For a (undirected) graph G, we define a set P_G that consists of all (ordered) pairs $(u, v) \in V(G) \times V(G)$ where u, v are connected by an edge of G. For a finite group Σ, a Σ-*labeled* graph is a pair (G, Λ) where G is graph and $\Lambda : P_G \to \Sigma$ is a function satisfying that $\Lambda(u, v) \times \Lambda(v, u)$ is equal to the identity of Σ for all $(u, v) \in P_G$. A *non-null cycle* in (G, Λ) is a cycle $(v_0, v_1, \ldots, v_m = v_0)$ in G satisfying that $\prod_{i=1}^{m} \Lambda(v_{i-1}, v_i)$ is a non-identity element of Σ. In the GFVS problem, the input is a Σ-labeled graph (G, Λ) for a finite group Σ and an integer k, and the goal is to determine whether there exists $X \subseteq V$ of size at most k such that $(G - X, \Lambda_{|P_{G-X}})$ contains no non-null cycles. We say a GFVS instance is *with non-identity labels* if Λ is required to satisfy $\Lambda(u, v) \neq \mathbf{1}$ for all $u \neq v$. A *consistent labeling* of a Σ-labeled graph (G, Λ) is a map $\mu : V \to \Sigma$ such that $\mu(v) = \mu(u) \cdot \Lambda(u, v)$ or equivalently $\mu(u) = \mu(v) \cdot \Lambda(v, u)$ for every edge $(u, v) \in E(G)$. It was known that containing no non-null cycle is equivalent to having a consistent labeling.

Lemma 4.10 ([28]). *A Σ-labeled graph has a consistent labeling if and only if it does not contain any non-null cycle.*

Note that the BIPARTIZATION problem is a special case of the GFVS problem with non-identity labels when $\Sigma = \mathbb{Z}_2$ and $\Lambda(u,v)$ is the non-identity element in $\mathbb{Z}_2$ for all $(u,v) \in P_G$. In this case, the consistent labeling simply corresponds to the assignment to the left or right part of the bipartition. Now we show how to generalize our BIPARTIZATION algorithm to the GFVS problem with non-identity labels. Suppose $|\Sigma| = g$. We can naturally associate every element of Σ to an index of $[g]$ by some bijection $\sigma : [g] \to \Sigma$. In this way, we can interpret, very similarly to what we did for BIPARTIZATION, a solution to a GFVS instance $(G = (V,E), \Lambda)$ as a map $\lambda : V \to \{0, 1, \ldots, g\}$, where $\lambda^{-1}(0)$ is the set $X \subseteq V$ of vertices to be removed and $\lambda^{-1}(i)$ for $i \in [g]$ corresponds to the set of vertices mapped to $\sigma(i)$ in a consistent labeling of $G - X$. Thus, we cam generalize the notion of configurations defined in the previous section. For a subset $V' \subseteq V$, a *configuration* of V' is a map $\lambda : V' \to \{0, 1, \ldots, g\}$. The configuration is Λ-*good* if for every $(u,v) \in E$ with $u, v \in V' \backslash \lambda^{-1}(0)$, we have $\lambda(v) = \lambda(u) \cdot \Lambda(u,v)$ or equivalently $\lambda(u) = \lambda(v) \cdot \Lambda(v,u)$. We say that two configurations $\lambda_1 : V_1 \to \{0, 1, \ldots, g\}$ and $\lambda_2 : V_2 \to \{0, 1, \ldots, g\}$ are *compatible* if they agree on $V_1 \cap V_2$.

Let $(G_{\mathcal{D}}, \Lambda)$ be the input Σ-labeled unit-disk graph where $G_{\mathcal{D}}$ is defined by a set $\mathcal{D}$ of n unit disks. By Lemma 4.10, our goal is to compute a subset $\mathcal{X} \subseteq \mathcal{D}$ of size at most k such that $G_{\mathcal{D}} - \mathcal{X}$ admits a consistent labeling. To adapt the algorithm, we need several things. First, like BIPARTIZATION, GFVS also admits a small candidate set: it is possible to find in polynomial time a set Cand $\subseteq \mathcal{D}$ of size $k^{O(g)}$ such that the potential solution $\mathcal{X}$ can be searched in Cand (see Lemma 5 of [6]). Second, we have to remark that a Σ-labeled graph with non-identity labels cannot admit a consistent labeling if it contains a clique of size $g+1$. Indeed it would mean that two adjacent vertices would be assigned to the same element of Σ which is only possible if the edge between these vertices is label with the identity.

With the two previous remarks, we can straightforwardly adapt the algorithm to this setting. The algorithm starts by applying Theorem 3.1 with $p = \lfloor \sqrt{k} \rfloor$ to obtain $\{\mathcal{D}_1, \ldots, \mathcal{D}_p\}$ a grid-respecting partition in polynomial time. Then it produces a set Cand $\subseteq \mathcal{D}$ of size $k^{O(g)}$ which contains the potential solution $\mathcal{X}$. Once again, it is possible to guess an index i satifying $|\mathcal{D}_i \cap \mathcal{X}| \leq k/p$ as well as the set $\mathcal{S} = \mathcal{D}_i \cap \mathcal{X}$ in at most $k^{O(\sqrt{k})}$ tries. Once this is done, we use a tree decomposition $\mathcal{T}$ of width $O(\sqrt{k})$ of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})$ guaranteed by Theorem 3.1 to solve the problem in $(k+g)^{O(g \cdot \sqrt{k})} n^{O(1)}$ time as follows.

Let V^* be the vertex set of $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})$ and $\pi : \mathcal{D} \to V^*$ be the quotient map of the edge contraction for obtaining $G_{\mathcal{D}}/(E^*_{\mathcal{D}} \cup E_{\mathcal{D}_i \backslash \mathcal{S}})$. The DP will store for every node $t \in \mathcal{T}$ and every Λ-good configuration α_t of $\pi^{-1}(\beta(t))$ the value of the minimum-cost Λ-good configuration of $\gamma(t)$ compatible with α_t. Again, every element x of $\beta(t)$ corresponds to either cell clique in $G_{\mathcal{D}}$ or a connected component of $\mathcal{D}_i \backslash \mathcal{S}$. In the first case, there are only $O(g+k)^{O(g)}$ Λ-good configurations for $\pi^{-1}(x)$, since at most g vertices in $\pi^{-1}(x)$ can have non-zero values in a Λ-good configuration of $\pi^{-1}(x)$. In the second case, there are only $g\Lambda$-good configurations for $\pi^{-1}(x)$ as it suffices to fix the value of one element to fix the rest by connectivity. Overall, since the size of $\beta(t)$ is $O(\sqrt{k})$, there is at most $(k+g)^{O(g \cdot \sqrt{k})}$ possible Λ-good configurations and thus we have the following result.

THEOREM 4.11. *There exists a randomized algorithm that solves the GFVS problem with non-identity labels on unit-disk graphs in $(k+g)^{O(g \cdot \sqrt{k})} n^{O(1)}$ time, where n is the number of vertices, k is the solution size, and g is the size of the group.*

5 Conclusion and Future Work

We prove a structural theorem for unit-disk graphs, which states that one can partition the vertices of a unit-disk graph $G_{\mathcal{D}}$ into p subsets $\mathcal{D}_1, \ldots, \mathcal{D}_p$ such that for any $i \in [p]$ and any $\mathcal{D}' \subseteq \mathcal{D}_i$, the graph $G_{\mathcal{D}}/(\mathcal{D}_i \backslash \mathcal{D}')$ admits a tree decomposition in which each bag consists of $O(p + |\mathcal{D}'|)$ cliques. This result can be viewed as an analog for unit-disk graphs of the "robust contraction decomposition" theorems for planar graphs and almost-embeddable graphs proved very recently by Marx et al. [42] and Bandyapadhyay et al. [6]. Our theorem finds both combinatorial and algorithmic applications. On the combinatorial side, we obtain the first CDT for unit-disk graphs, resolving an open question in the work by [48]. On the algorithmic side, our theorem yields a new parameterized algorithm for bipartization on unit-disk graphs, which runs in $2^{O(\sqrt{k} \log k)} \cdot n^{O(1)}$ time, where k denotes the solution size. Our algorithm significantly improves the previous slightly subexponential-time parameterized algorithm given by [40] which runs in $2^{O(k^{27/28})} \cdot n^{O(1)}$ time. We also give a $2^{\Omega(\sqrt{k})} \cdot n^{O(1)}$-time lower bound for the problem based on the ETH, which implies that our algorithm is almost optimal.

Next, we raise some open questions for future study. The first question is whether we can extend our structural theorem to more general graph classes. An interesting case is the class of (general) disk graphs, which generalizes both planar graphs and unit-disk graphs. As this type of structural theorem holds for both planar graphs and unit-disk graphs, it is natural to ask whether one can obtain similar results for disk graphs. The second question is to improve our CDT for unit-disk graphs. In Theorem 4.1, the treewidth bound we have is $O(p^2)$, while a bound of $O(p)$ was known for planar graphs. Therefore, it is interesting to ask whether one can prove a CDT for unit-disk graphs with a subquadratic (or even linear) treewidth bound, or prove a quadratic lower bound for the treewidth. Finally, we want to ask whether the general GFVS problem (possibly with identity labels) also admits a subexponential parameterized algorithm.

References

[1] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. 2005. $o(\sqrt{\log n})$ approximation algorithms for min Uncut, min 2CNF deletion, and directed cut problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*. 573–581.

[2] Jochen Alber, Henning Fernau, and Rolf Niedermeier. 2003. Graph separators: A parameterized view. *J. Comput. Syst. Sci.* 67, 4 (2003), 808–832.

[3] Jochen Alber and Jirí Fiala. 2004. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *J. Algorithms* 52, 2 (2004), 134–151.

[4] Shinwoo An and Eunjin Oh. 2021. Feedback vertex set on geometric intersection graphs. In *Proceedings of the 32nd International Symposium on Algorithms and Computation (ISAAC '21)*. Hee-Kap Ahn and Kunihiko Sadakane (Eds.), LIPIcs, Vol. 212, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 47:1–47:12.

[5] Aistis Atminas and Viktor Zamaraev. 2018. On forbidden induced subgraphs for unit disk graphs. *Discrete Comput. Geom.* 60, 1 (2018), 57–97.

[6] Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. 2022. Subexponential parameterized algorithms for cut and cycle hitting problems on H-minor-free graphs. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '22)*. 2063–2084.

[7] Thang Nguyen Bui and Andrew Peck. 1992. Partitioning planar graphs. *SIAM J. Comput.* 21, 2 (1992), 203–215.

[8] Sergio Cabello and Miha Jejcic. 2015. Shortest paths in intersection graphs of unit disks. *Comput. Geom.* 48, 4 (2015), 360–367.

[9] Timothy M. Chan and Dimitrios Skrepetos. 2017. All-pairs shortest paths in geometric intersection graphs. In *Proceedings of the Workshop on Algorithms and Data Structures (WADS '17)*. Springer, 253–264.

[10] Timothy M. Chan and Dimitrios Skrepetos. 2019. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *J. Comput. Geom.* 10, 2 (2019), 3–20.

[11] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. 1990. Unit disk graphs. *Discrete Math.* 86, 1–3 (1990), 165–177.

[12] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.

[13] Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. 2018. A framework for ETH-tight algorithms and lower bounds in geometric intersection graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC '18)*. 574–586.

[14] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. 2005. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM* 52, 6 (2005), 866–893.

[15] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-Ichi Kawarabayashi. 2011. Contraction decomposition in H-minor-free graphs and algorithmic applications. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC '11)*. 441–450.

[16] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. 2010. Approximation algorithms via contraction decomposition. *Combinatorica* 30, 5, (2010), 533–552.

[17] Frederic Dorn, Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. 2013. Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs. *Inf. Comput.* 233 (2013), 60–70.

[18] Samuel Fiorini, Nadia Hardy, Bruce Reed, and Adrian Vetta. 2008. Planar graph bipartization in linear time. *Discrete Appl. Math.* 156, 7 (2008), 1175–1180.

[19] Fedor V. Fomin, Daniel Lokshtanov, Sudeshna Kolay, Fahad Panolan, and Saket Saurabh. 2020. Subexponential algorithms for rectilinear Steiner tree and arborescence problems. *ACM Trans. Algorithms*, 16, 2 (2020), 21:1–21:37.

[20] Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. (2016). Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In *Proceedings of the IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS '16)*. Irit Dinur (Ed.), Hyatt Regency, IEEE Computer Society, 515–524.

[21] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. 2019. Decomposition of map graphs with applications. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP '19)*. Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (Eds.), LIPIcs, Vol. 132, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 60:1–60:15.

[22] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. 2019. Finding, hitting and packing cycles in subexponential time on unit disk graphs. *Discret. Comput. Geom.* 62, 4 (2019), 879–911.

[23] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. 2020. ETH-tight algorithms for long path and cycle on unit disk graphs. In *Proceedings of the 36th International Symposium on Computational Geometry (SoCG '20)*. Sergio Cabello and Danny Z. Chen (Eds.), LIPIcs, Vol. 164, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 44:1–44:18.

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

[25] Jie Gao and Li Zhang. 2005. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM J. Comput.* 35, 1 (2005), 151–169.

[26] Michel X. Goemans and David P. Williamson. 1998. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica* 18, 1 (1998), 37–59.

[27] Alexander Grigoriev and Hans L. Bodlaender. 2007. Algorithms for graphs embeddable with few crossings per edge. *Algorithmica* 49, 1 (2007), 1–11.

[28] Sylvain Guillemot. 2011. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optim.* 8, 1 (2011), 61–71.

[29] MohammadTaghi Hajiaghayi. 2016. Contraction and minor graph decomposition and their algorithmic applications. *Filmed Talk at Microsoft Research.*

[30] William K. Hale. 1980. Frequency assignment: Theory and applications. *Proc. IEEE* 68, 12 (1980), 1497–1514.

[31] Falk Hüffner. 2009. Algorithm engineering for optimal graph bipartization. *J. Graph Algorithms Appl.* 13, 2 (2009), 77–98.

[32] Bart M. P. Jansen, Marcin Pilipczuk, and Erik Jan van Leeuwen. A deterministic polynomial kernel for odd cycle transversal and vertex multiway cut in planar graphs. In *Proceedings of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS '19)*. Rolf Niedermeier and Christophe Paul (Eds.), LIPIcs, Vol. 126, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 39:1–39:18 .

[33] Ken-Ichi Kawarabayashi and Bruce Reed. 2010. An (almost) linear time algorithm for odd cycles transversal. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '10)*. SIAM, 365–378.

[34] Philip N. Klein. 2006. A subset spanner for planar graphs, with application to subset TSP. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC '06)*. 749–756.

[35] Philip N. Klein. 2008. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.* 37, 6 (2008), 1926–1952.

[36] Philip N. Klein and Dániel Marx. 2012. Solving planar k-terminal cut in $O(n^{c\sqrt{k}})$ time. In *Proceedings of the 39th International Colloquium Automata, Languages, and Programming (ICALP '12)*. Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer (Eds.), Lecture Notes in Computer Science, Vol. 7391, Springer, 569–580.

[37] Philip N. Klein and Dániel Marx. 2014. A subexponential parameterized algorithm for subset TSP on planar graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '14*). Chandra Chekuri, editor, SIAM, 1812–1830.

[38] Stefan Kratsch and Magnus Wahlström. 2014. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms* 10, 4 (2014), 20:1–20:15.

[39] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. 2014. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms (TALG)* 11, 2 (2014), 1–31.

[40] Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. 2022. Subexponential parameterized algorithms on disk graphs. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '22).* 2005–2031.

[41] Daniel Lokshtanov, Saket Saurabh, and Magnus Wahlström. 2012. Subexponential parameterized odd cycle transversal on planar graphs. In *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '12).* Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan (Eds.), LIPIcs, Vol. 18, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 424–434.

[42] Dániel Marx, Pranabendu Misra, Daniel Neuen, and Prafullkumar Tale. 2022. A framework for parameterized subexponential algorithms for generalized cycle hitting problems on planar graphs. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '22).* 2085–2127.

[43] Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. 2018. On subexponential parameterized algorithms for steiner tree and directed subset TSP on planar graphs. In *Proccedings of the 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS '18).* Mikkel Thorup (Ed.), IEEE Computer Society, 474–484.

[44] Dániel Marx and Michal Pilipczuk. 2015. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In *Proccedings of the Algorithms 23rd Annual European Symposium (ESA '15).* Nikhil Bansal and Irene Finocchi (Eds.), Lecture Notes in Computer Science, Vol. 9294, Springer, 865–877.

[45] Jesper Nederlof. 2020. Detecting and counting small patterns in planar graphs in subexponential parameterized time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '20).* Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy (Eds.), ACM, 1293–1306.

[46] Emily Palmer. The hidden algorithm of Ore's theorem on Hamiltonian cycles. *Comput. Math. Appl.* 34, 11 (1997), 113–119.

[47] Fahad Panolan, Saket Saurabh, and Meirav Zehavi. 2019. Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '19).* Timothy M. Chan. (Ed.), SIAM, 1035–1054.

[48] Fahad Panolan, Saket Saurabh, and Meirav Zehavi. 2019. Parameterized computational geometry via decomposition theorems. In *Proceedings of the 13th International Conference Algorithms and Computation (WALCOM '19).* Gautam K. Das, Partha S. Mandal, Krishnendu Mukhopadhyaya, and Shin-Ichi Nakano. (Eds.), Lecture Notes in Computer Science, Vol. 11355, Springer, 15–27.

[49] Bruce Reed, Kaleigh Smith, and Adrian Vetta. 2004. Finding odd cycle transversals. *Oper. Res. Lett.* 32, 4 (2004), 299–301.

[50] Siamak Tazari. 2012. Faster approximation schemes and parameterized algorithms on (odd-)H-minor-free graphs. *Theor. Comput. Sci.* 417 (2012), 95–107.

[51] Haitao Wang and Jie Xue. 2020. Near-optimal algorithms for shortest paths in weighted unit-disk graphs. *Discrete Comput. Geom.* 64, 4 (2020), 1141–1166.

[52] Mihalis Yannakakis. 1978. Node-and edge-deletion NP-complete problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computingcomputing (STOC '78).* 253–264.

[53] Yu-Shuan Yeh, Joanne C. Wilson, and Stuart C. Schwartz. 1984. Outage probability in mobile telephony with directive antennas and macrodiversity. *IEEE Trans. Veh. Technol.* 33, 3 (1984), 123–127.

Received 11 December 2022; revised 10 March 2024; accepted 23 March 2024