ETH-Tight Algorithm for Cycle Packing on Unit Disk Graphs

Shinwoo An ✉
POSTECH, Pohang, South Korea

Eunjin Oh ✉
POSTECH, Pohang, South Korea

Abstract

In this paper, we consider the Cycle Packing problem on a unit disk graph defined as follows. Given a unit disk graph G with n vertices and an integer k, the goal is to find a set of k vertex-disjoint cycles of G if it exists. Our algorithm runs in time $2^{O(\sqrt{k})}n^{O(1)}$. This improves the $2^{O(\sqrt{k}\log k)}n^{O(1)}$-time algorithm by Fomin et al. [SODA 2012, ICALP 2017]. Moreover, our algorithm is optimal assuming the exponential-time hypothesis.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Unit disk graphs, cycle packing, tree decomposition, parameterized algorithm

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.7

Related Version *Full Version*: `https://arxiv.org/abs/2403.11426`

Funding This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (No.RS-2023-00209069).

1 Introduction

The Cycle Packing problem is a fundamental graph problem defined as follows. Given an undirected graph G and an integer k, the goal is to check if there is a set of k vertex-disjoint cycles of G. This problem is NP-hard even for planar graphs. This motivates the study from the viewpoints of parameterized algorithms [29] and approximation algorithms [20, 28]. For approximation algorithms, we wish to approximate the maximum number of vertex-disjoint cycles of G in polynomial time. The best known polynomial time algorithm has approximation factor of $O(\sqrt{\log n})$ [28]. This is almost optimal in the sense that it is quasi-NP-hard to approximate the maximum number of vertex-disjoint cycles of a graph within a factor of $O(\log^{1/2-\epsilon} n)$ for any $\epsilon > 0$ [28]. Several variants also have been considered, for instance, finding a maximum number of vertex-disjoint triangles [22], finding a maximum number of vertex-disjoint odd cycles [26], finding a maximum number of edge-disjoint cycles [20], and finding a maximum number of vertex-disjoint cycles in directed graphs [20].

In this paper, we study the Cycle Packing problem from the viewpoint of parameterized algorithms when the parameter k is the number of vertex-disjoint cycles. This problem is one of the first problems studied from the perspective of Parameterized Complexity. By combining the Erdős-Pósa theorem [13] with a $2^{O(tw\log tw)}n^{O(1)}$-time standard dynamic programming algorithm for this problem, where tw is the treewidth of the input graph, one can solve the Cycle Packing problem in $2^{O(k\log^2 k)}n^{O(1)}$ time. This algorithm was improved recently by Lokshtanov et al. [29]. They improved the exponent on the running time by a factor of $O(\log\log k)$. As a lower bound, no algorithm for the Cycle Packing problem runs in $2^{o(tw\log tw)}n^{O(1)}$ time assuming the exponential-time hypothesis (ETH) [8].

For several classes of graphs, the Cycle Packing problem can be solved significantly faster. For planar graphs, Bodlaender et al. [4] presented a $2^{O(\sqrt{k})}n^{O(1)}$-time algorithm, and showed that the Cycle Packing problem admits a linear kernel. Also, one can obtain an algorithm with the same time bound using the framework of Dorn et al. [12]. Later, Dorn et al. [11] presented a $2^{O(\sqrt{k})}n^{O(1)}$-time algorithm which works on H-minor-free graphs for a fixed graph H. As a main ingredient, they presented a branch decomposition of a H-minor-free graph which has a certain structure called the *Catalan structure*. This structure allows them to bound the ways a cycle may cross a cycle separator of an H-minor-free graph. Also, there is a subexponential-time parameterized algorithm for the Cycle Packing problem for map graphs [17]. These results raise a natural question whether a subexponential-time algorithm for the Cycle Packing problem can be obtained for other graph classes.

In this paper, we focus on *unit disk graphs*. For a set V of points in the plane, the *unit disk graph* G is defined as the undirected graph whose vertices correspond to the points of V such that two vertices are connected by an edge in G if and only if their Euclidean distance is at most one. It can be used as a model for broadcast networks: The points of V represent transmitter-receiver stations with the same transmission power. Unit disk graphs have been studied extensively for various algorithmic problems [5, 7, 14, 23, 24, 25]. Also, several NP-complete problems have been studied for unit disk graphs (and geometric intersection graphs) from the viewpoint of parameterized algorithms, for example, the Steiner Tree, Bipartization, Feedback Vertex Set, Clique, Vertex Cover, Long Path and Cycle Packing problems [1, 2, 3, 6, 15, 16, 19, 30, 31]. All problems listed above, except for the Steiner Tree problem, admit subexponential-time parameterized algorithms for unit disk graphs. The study of parameterized algorithms for unit disk graphs and geometric intersection graphs is currently a highly active research area in Computational Geometry.

To the best of our knowledge, the study of subexponential-time *parameterized algorithms* for unit disk graphs was initiated by Fomin et al. [19]. They focused on the Feedback Vertex Set[1] and Cycle Packing problems and presented $2^{O(k^{0.75}\log k)}n^{O(1)}$-time algorithms. Later, they were improved to take $2^{O(\sqrt{k}\log k)}n^{O(1)}$ time by [16], and this approach also works for the Long Path and Long Cycle problems[2] with the same time bound. Since the best known lower bound for all these problems is $2^{\Omega(\sqrt{k})}n^{O(1)}$ assuming ETH [16], it is natural to ask if these problems admit ETH-tight algorithms. Recently, this question was answered affirmatively for all problems mentioned above, except for the Cycle Packing problem [1, 18]. It seems that the approaches used in [1, 18] are not sufficient for obtaining a faster algorithm for the Cycle Packing problem. After preprocessing, they reduce the original problems for unit disk graphs to the weighted variants of the problems for graphs with treewidth $O(\sqrt{k})$, and then they use $2^{O(tw)}n^{O(1)}$-time algorithms for the weighted variants of the problems for a graph with treewidth tw. However, no algorithm for the Cycle Packing problem runs in $2^{o(tw\log tw)}n^{O(1)}$ time for a graph with treewidth tw assuming ETH [8].

Our Result. In this paper, we present an ETH-tight parameterized algorithm for the Cycle Packing problem on unit disk graphs with n vertices, which runs in $2^{O(\sqrt{k})}n^{O(1)}$ time. No ETH-tight algorithm even for the non-parameterized version was known prior to this work. In the case of the Long Path/Cycle and Feedback Vertex Set problems, ETH-tight algorithms running in $2^{O(\sqrt{n})}$ time were already known [9] before the ETH-tight

[1] Given a graph G and an integer k, find a set F of k vertices such that $G - F$ does not have a cycle.
[2] Given a graph G and an integer k, find a path and a cycle of G with k vertices, respectively.

parameterized algorithms were presented [1, 18]. As a tool, we introduce a new recursive decomposition of the plane into regions with $O(1)$ boundary components with respect to a unit disk graph G such that the edges of G crossing the boundary of each region form a small number of cliques. It can be used for other problems such as the non-parameterized version of the Odd Cycle Packing problem, and the parameterized versions of the d-Cycle Packing and 2-Bounded-Degree Vertex Deletion problems on unit disk graphs. Also, although we describe our results for unit disk graphs in the main text, they work for more general graph classes such as the intersection graphs of similarly-sized disks and squares. All missing proofs and omitted details can be found in the full version.

2 Overview of Our Algorithm

In this section, we give an overview of our algorithm for the Cycle Packing problem on unit disk graphs. We are given a unit disk graph $G = (V, E)$ along with its geometric representation. Here, each edge of G is drawn as a line segment connecting its endpoints. We do not distinguish a vertex of G and its corresponding point of $\mathbb{R}^2$ where it lies. Let $\mathcal{M}$ be a partition of the plane into interior-disjoint squares of diameter one. We call it a *map* of G,[3] and a square of $\mathcal{M}$ a *cell* of $\mathcal{M}$. Notice that the subgraph of G induced by $M \cap V$ is a clique. Throughout this paper, we let α be the maximum number of cells of $\mathcal{M}$ intersected by one edge of G. Note that $\alpha = O(1)$. For any integer r, a cell M of $\mathcal{M}$ is called an *r-neighboring cell* of M' of $\mathcal{M}$ if a line segment connecting M and M' intersects at most r cells of $\mathcal{M}$. Note that a cell of $\mathcal{M}$ has $O(1)$ α-neighboring cells.

For a region $A \subseteq \mathbb{R}^2$, we use ∂A to denote the boundary of A. For a subset U of V, we let $G[U]$ be the subgraph of G induced by U. For convenience, we let $G[V \setminus U] = G \setminus U$. We often use $V(G)$ and $E(G)$ to denote the vertex set of G and the edge set of G, respectively. For a set Γ' of paths and cycles of G, we let $\mathsf{end}(\Gamma')$ be the set of end points of the *paths* of Γ'. Moreover, we may seen Γ' as a set of edges, and we let $V(\Gamma')$ be the set of all vertices of the *paths and cycles* of Γ'.

2.1 Standard Approach and Main Obstacles

Let Γ be a set of k vertex-disjoint cycles of G. Ideally, we want to recursively decompose the plane into smaller regions each consisting of $O(1)$ boundary curves so that "very few" edges of Γ cross the boundary of each region. Then we want to compute Γ using dynamic programming on the recursive decomposition of the plane. For each region A, we want to guess the set E' of edges of Γ crossing the boundary of A, and then want to guess the pairing $\mathcal{P}$ of the vertices of $\mathsf{end}(E') \cap A$ such that two vertices of $\mathsf{end}(E') \cap A$ belong to the same pair of $\mathcal{P}$ if and only if a path of $\Gamma \setminus E'$ has them as its endpoints. Then for a fixed pair $(E', \mathcal{P})$, it suffices to compute the maximum number of vertex-disjoint cycles of Γ' contained in A over all sets Γ' of vertex-disjoint cycles and paths of the subgraph of G induced by the edges fully contained in A and the edges of E' which match the given information. In fact, this is a standard way to deal with this kind of problems.

The Cycle Packing problem on planar graphs can be solved using this approach [12]. Given a simple closed curve η intersecting a planar graph only at its vertices, consider the maximal paths of the cycles of Γ contained in the interior of η. Clearly, they do not intersect in its drawing, and thus they have a Catalan structure. Thus once the crossing points of

[3] It is a grid in the case of unit disk graphs, but we define a map as general as possible in the full version.

the cycles of Γ with η are fixed, we can enumerate $2^{O(w)}$ pairings one of which is the correct pairing of the endpoints of the paths, where w is the number of crossing points between Γ and η. However, we cannot directly apply this approach to unit disk graphs mainly because G has a large clique, and two vertex-disjoint cycles of G cross in their drawings. More specifically, we have the following three obstacles.

First issue. We do not have any tool for recursively decomposing the plane into smaller regions such that the boundary of each region is crossed by a small number of edges of Γ. De Berg et al. [9] presented an algorithm for computing a rectangle separating V in a balanced way such that the total *clique-weight* of the cells of $\mathcal{M}$ crossing the boundary of the rectangle is $O(\sqrt{n})$, where the clique-weight of a cell M is defined as $\log(|M \cap V| + 1)$. Using this, we can show that the boundary of this rectangle is crossed by $O(\sqrt{n})$ edges of Γ. However, this only works for a single recursion step. If we apply the algorithm by de Berg et al. [9] recursively, a region we obtained after d steps can have $\Theta(d)$ boundary components. Note that $d = \omega(1)$ in the worst case. To handle this, whenever the number of boundary components of a current region exceeds a certain constant, we have to reduce the number of boundary components using a balanced separator separating the boundary components. However, it seems unclear if it is doable using the algorithm by de Berg et al. [9] as their separator works for fat objects, but the boundary components are not necessarily fat. Another issue is that we need a separator of complexity $O(\sqrt{k})$ instead of $O(\sqrt{n})$.

Second issue. Even if we have a recursive decomposition of the plane into regions such that the boundary of each region A is crossed by $O(\sqrt{k})$ edges of Γ, we have to guess the number of edges of Γ crossing ∂A among all edges of G crossing ∂A. Since the number of edges of G crossing ∂A can be $\Theta(n)$, a naive approach gives $n^{O(\sqrt{k})}$ candidates for the correct guess. This issue happens also for other problems on unit disk graphs such as Long Path/Cycle and Feedback Vertex Set. The previous results on these problems [1, 18] handle this issue by using the concept of the clique-weight of a cell of $\mathcal{M}$, which was introduced by de Berg et al. [9]. We can handle this issue as they did.

Third issue. Suppose that we have a recursive decomposition with desired properties, and we have the set E' of $O(\sqrt{k})$ edges of Γ crossing the boundary of A for each region A. For convenience, assume that ∂A is connected. The number of all pairings of $\mathsf{end}(E') \cap \partial A$ is $2^{O(\sqrt{k}\log k)}$, which exceeds the desired bound. But not all such pairings can be the correct pairing of a set of maximum-number of cycles of G. For a vertex a of $\mathsf{end}(E') \cap \partial A$, let $\bar{a}$ be the first point on A from a along the edge of E' incident to a. If no two cycles of Γ cross in their drawings, the cyclic order of $\bar{a}, \bar{b}, \bar{a}'$ and $\bar{b}'$ along ∂A is either $\langle \bar{a}, \bar{b}, \bar{a}', \bar{b}' \rangle$ or $\langle \bar{b}, \bar{a}, \bar{a}', \bar{b}' \rangle$ for any two pairs (a, b) and (a', b') of $\mathcal{P}$, where $\mathcal{P}$ denotes the correct pairing of $\mathsf{end}(E') \cap A$. That is, $\mathcal{P}$ is a *non-crossing pairing*. It is known that for a fixed set P (which is indeed $\mathsf{end}(E') \cap A$), the number of non-crossing pairings of P is $2^{O(|P|)}$ (which is indeed $2^{O(\sqrt{k})}$). However, two cycles of Γ can cross in general, and thus $\mathcal{P}$ is not necessarily non-crossing.

2.2 Our Methods

We can handle the three issues using the following two main ideas.

Surface cut decomposition of small clique-weighted width. We handle the first issue by introducing a new decomposition for unit disk graphs, which we call a *surface cut decomposition*. It is a recursive decomposition of the plane into regions (called *pieces*) with

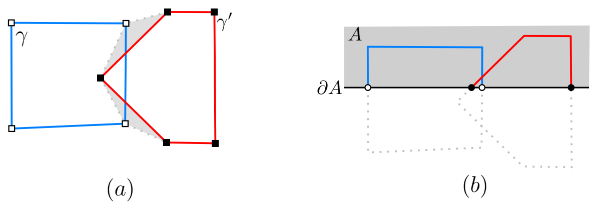


Figure 1 (a) The drawings of two cycles of an optimal solution may cross. We cannot replace them into triangles (gray color). (b) Two paths of $\bar{\Pi}$ are cross-ordered, and their drawings cross.

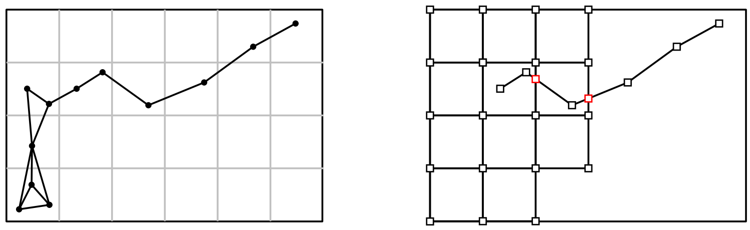
$O(1)$ boundary components such that each piece has *clique-weight* $O(\sqrt{\ell})$, where ℓ denotes the number of vertices of degree at least three in G. Let $\mathsf{cut}(A)$ denote the set of edges with at least one endpoint on A that intersect ∂A. An edge of $\mathsf{cut}(A)$ might have both endpoints in A. For an illustration, see Figure 3. We show that the total clique-weight of the cells of $\mathcal{M}$ containing the endpoints of the edges of $\mathsf{cut}(A)$ is small for each piece A. The clique-weight of A is defined as the total clique-weight of the cells containing the endpoints of $\mathsf{cut}(A)$. Recall that the clique-weight of a cell M is defined as $\log(|M \cap V| + 1)$.

To handle the second issue, we show that $O(1)$ vertices contained in M lie on the cycles of Γ' for each cell M, and every cycle of $\Gamma \setminus \Gamma'$ is a triangle, where Γ' is the set of cycles of Γ visiting at least two vertices from different cells of $\mathcal{M}$. We call this property the *bounded packedness property.* Assume that we have a cell M of clique-weight ω. It is sufficient to specify the vertices in M appearing in Γ'. For the other vertices in M, we construct a maximum number of triangles. By the bounded packedness property of Γ, the number of choices of the edges in M appearing in Γ' is reduced to $|M \cap V|^{O(1)} = 2^{O(|\log(|M \cap V|+1)|} = 2^{O(\omega)}$. This also implies that the boundary of each region A is crossed by $O(\omega)$ cut edges.

Deep analysis on the intersection graphs of cycles. We handle the third issue as follows. We choose Γ in such a way that it has the minimum number of edges among all possible solutions. Consider a piece A not containing a hole. Suppose that we have the set E' of $O(\sqrt{k})$ cut edges of Γ crossing the boundary of A. Due to the bounded packedness property, we can ignore the cycles of Γ fully contained in a single cell of $\mathcal{M}$. Let Γ' be the set of the remaining cycles of Γ. Let Π be the set of path components of the subgraph of Γ' induced by $V \cap A$. An endpoint of a path π of Π is incident to a cut edge of A along Γ. We extend the endpoint of π along the cut edge until it hits ∂A. Let $\bar{\Pi}$ be the set of the resulting paths.

We aim to compute a small number of pairings of $\mathsf{end}(\bar{\Pi})$ one of which is the correct pairing of $\bar{\Pi}$. To do this, we consider the intersection graph $\mathcal{G}$ of the paths of $\bar{\Pi}$: a vertex of $\mathcal{G}$ corresponds to a path of $\bar{\Pi}$, and two vertices are adjacent in $\mathcal{G}$ if and only if their paths cross in their drawings. We show that the intersection graph $\mathcal{G}$ of $\bar{\Pi}$ is $K_{z,z}$-free for a constant z. We call this property the *quasi-planar property.* Note that the paths of $\bar{\Pi}$ may cross even if Γ has the bounded packedness property. See Figure 1.

Then we relate the paring of $\mathsf{end}(\bar{\Pi})$ to $\mathcal{G}$. We say two paths of $\bar{\Pi}$, one ending at a and b, and one ending at a' and b', are *cross-ordered*, if a, a', b and b' appear along ∂A in this order. For any two *cross-ordered* paths of $\bar{\Pi}$, their drawings cross since they are contained in A. Given a pairing $\mathcal{P}$ of $\mathsf{end}(\bar{\Pi})$, we define the *circular arc crossing graph* of $\mathcal{P}$ such that each vertex corresponds to a pair of $\mathcal{P}$, and two vertices are adjacent if their corresponding pairs are cross-ordered. By the previous observation, the circular arc crossing graph is isomorphic to a subgraph of $\mathcal{G}$. Thus it suffices to enumerate all pairings whose corresponding circular arc crossing graphs are $K_{z,z}$-free. We show that the number of all such pairings is $2^{O(\omega)}$.



(a) (b)

Figure 2 (a) Illustration of G and $\mathcal{M}$. (b) The base vertices are marked with black boxes, and the cross vertices are marked with red boxes.

In this way, we can enumerate $2^{O(\sqrt{k})}$ pairings of $\mathsf{end}(\bar{\Pi})$ one of which is the correct pairing of $\bar{\Pi}$ in the case that ∂A is connected. We can handle the general case where ∂A has more than one curve in a similar manner although there are some technical issues.

3 Surface Cut Decomposition

In this section, we present an algorithm for computing a surface cut decomposition of clique-weighted width $O(\sqrt{\ell})$, where ℓ is the number of vertices of degree at least three in G. We first decompose the plane into smaller pieces with respect to the *map sparsifier* H of G, and then we perturb the boundaries of the pieces so that each piece has $O(\sqrt{\ell})$ clique-weighted width. The *map sparsifier* is defined as follows. See Figure 2. Consider all α-neighboring cells of the cells containing vertices of G of degree at least three. Let H' be the plane graph consisting of all boundary edges of such cells. In addition to them, we add the edges of G whose both endpoints have degree at most two to H'. The vertices of H' are called the *base vertices*. Note that the number of base vertices contained in a single cell is $O(1)$ because otherwise, these base vertices should have degree at least three by the definition of the cells. Two edges of H' can cross. In this case, we add such a crossing point as a vertex of H' and split the two edges with respect to the new vertex. These vertices are called the *cross vertices*. Let H be the resulting planar graph, and we call it a *map sparsifier* of G with respect to $\mathcal{M}$.

We define a *surface decomposition* of planar graphs and a *surface cut decomposition* of unit disk graphs. A region A is *regular closed* if the closure of the interior of A is A itself. we say a regular closed and interior-connected region A is an *s-piece* if $\mathbb{R}^2 \setminus A$ has s connected components. If $s = O(1)$, we call A a *piece*,[4] and we call s the *rank* of A. See Figure 3(a). The boundary of a component of $\mathbb{R}^2 \setminus A$ is a closed curve. We call it a *boundary curve* of A.

Surface decomposition. A *surface decomposition* of a plane graph H with vertex weights $c : V(H) \to \mathbb{R}^+$ is a pair $(T, \mathcal{A})$ where T is a rooted binary tree, and $\mathcal{A}$ maps a node t of T into a piece A_t satisfying **(A1–A3)**. For a node t and two children t', t'' of t,

- **(A1)** ∂A_t intersects the planar drawing of H only at its vertices,
- **(A2)** $A_{t'}, A_{t''}$ are interior-disjoint and $A_{t'} \cup A_{t''} = A_t$, and
- **(A3)** $|V(H) \cap A_t| \leq 2$ if t is a leaf node of T.

The *weight* of a node t is defined as the sum of weights of the vertices of $V(H)$ lying on ∂A_t. The *weighted width* of $(T, \mathcal{A})$ is defined as the maximum weight of the nodes of T. A surface decomposition can be considered as the weighted variant of the *structured recursive*

[4] A piece is a planar surface, and this is why we call the resulting decomposition a surface decomposition.

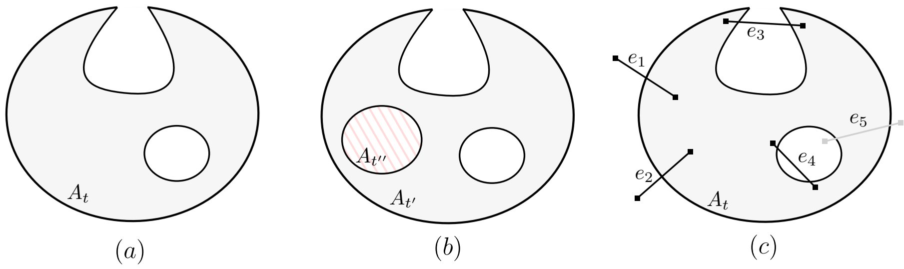


Figure 3 (a) A planar surface with rank 2. (b) Two regions $A_{t'}$ and $A_{t''}$ partition A_t. (c) The sold edges (e_1, e_2, e_3 and e_4) are cut edges, and e_5 is not a cut edge of A_t.

separator decomposition of a planar graph introduced by [27]. Our strategy is similar to theirs, but one difference is that we need a balanced cycle separator consisting of vertices whose total *weight* is small. To do this, we generalize the results of [10] and of [21]. More specifically, Djidjev [10] showed that H has a 2/3-balanced separator with the desired weight, but the separator is not necessarily a cycle. On the other hand, Har-Peled and Nayyeri [21] showed that H has a 2/3-balanced cycle separator of the desired weight in the case that all vertex weights of H are the same.

▶ **Theorem 1.** *For a plane graph $H = (V, E)$ with vertex weight $c(\cdot)$ with $1 \leq c(v) \leq n^{O(1)}$ for all $v \in V$, one can compute a surface decomposition of weighted width $O(\sqrt{\sum_{v \in V}(c(v))^2})$ in $O(n \log n)$ time, where n denotes the number of vertices of H.*

Surface cut decomposition. A *surface cut decomposition*, sc-decomposition in short, of G is a pair $(T, \mathcal{A})$ where T is a rooted binary tree, and $\mathcal{A}$ maps a node t of T into a piece A_t satisfying **(C1–C4)**. For a node t of T and two children t', t'' and a cell M of $\mathcal{M}$,

- **(C1)** $V(G) \cap \partial A_t = \emptyset$,
- **(C2)** $A_{t'}, A_{t''}$ are interior-disjoint, $A_{t'} \cup A_{t''} = A_t$,
- **(C3)** $V(G) \cap A_t$ is contained in the union of at most two cells of $\mathcal{M}$ for a leaf node t, and
- **(C4)** there are $O(1)$ leaf nodes of T containing points of $M \cap V(G)$ in their pieces.

Let $\mathsf{cut}(t) = \mathsf{cut}(A_t)$ be the set of edges of G with at least one endpoint in A_t that intersect ∂A_t. See Figure 3(c). The *clique-weighted width* of a node t is defined as the sum of the clique-weights of all cells of $\mathcal{M}$ containing the endpoints of the edges of $\mathsf{cut}(t)$. The *clique-weighted width* of an sc-decomposition $(T, \mathcal{A})$ is defined as the maximum clique-weight of the nodes of T.

▶ **Theorem 2.** *Given a unit disk graph G along with its geometric representation, one can compute an sc-decomposition of clique-weighted width $O(\sqrt{\ell})$ in polynomial time, where ℓ is the number of vertices of degree at least three in G.*

Sketch of the proof. Let H be the map sparsifier of G. For a vertex v of H, we set its weight as the sum of the clique-weights of all 2α-neighboring cells of M_v in $\mathcal{M}$, where M_v is the cell of $\mathcal{M}$ containing v. This will ensure that the clique-weighted width of the surface cut decomposition of G constructed from a surface decomposition $(T, \mathcal{A})$ of H is at most the weight of $(T, \mathcal{A})$.

Since the desired clique-weight of each piece is $O(\sqrt{\ell})$, not $O(\sqrt{n})$, we first handle the vertices of H of degree less than three. Let H_3 be the minor of H obtained by contracting each maximal path consisting of degree-1 and degree-2 vertices and then by removing all degree-1 vertices. Then the maximum weight of each vertex of H (and H_3) is at most $O(\log \ell)$, and the sum of the squared weights of all vertices of H_3 is $O(\ell)$. Then we construct a surface decomposition of H_3 of weighted width $O(\sqrt{\ell})$ using Theorem 1. Then we modify it to obtain a surface decomposition of H of weighted width $O(\sqrt{\ell})$ by subdividing each piece of $\mathcal{A}$ further to handle the vertices of $H \setminus V(H_3)$.

Finally, we compute an sc-decomposition $(T, \bar{\mathcal{A}})$ of G from the surface decomposition $(T, \mathcal{A})$ of H. The pieces of $\mathcal{A}$ are constructed with respect to the vertices of H. We perturb the boundaries of the pieces of $\mathcal{A}$ slightly so that no resulting piece $\bar{A}$ contains a vertex of G on its boundary. We can do this without changing the topological structures of the pieces. After doing this, all conditions **(C1–C4)** are satisfied.

Now we show that the clique-weighted width of $(T, \bar{\mathcal{A}})$ is $O(\sqrt{\ell})$. In particular, we show that for each node t of T, the total clique-weight of the cells of $\mathcal{M}$ containing the endpoints of the edges of $\mathsf{cut}(t)$ is $O(\sqrt{\ell})$. Let $\mathsf{bd}(t)$ denote the set of vertices of $V(H) \cap \partial A_t$. The weight of t in the surface decomposition of H is defined as the total weight of $\mathsf{bd}(t)$. Also, the weight $c_H(v)$ of v in H is defined as the total clique-weight of the cells of $\mathcal{M}_{\mathsf{nb}}(v)$, where $\mathcal{M}_{\mathsf{nb}}(v)$ denotes the set of all α-neighboring cells of M_v. Therefore, it suffices to show that the endpoints of $\mathsf{cut}(t)$ are contained in the cells of the union of $\mathcal{M}_{\mathsf{nb}}(v)$ for all vertices $v \in \mathsf{bd}(t)$.

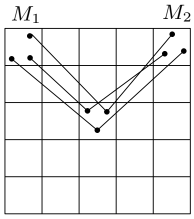
Let $e = ab$ be an edge of $\mathsf{cut}(t)$, and let M be a cell containing a point of $e \cap \partial \bar{A}_t$. Consider the case that either M_a or M_b contains a vertex of degree at least three. Here, M_a and M_b are the cells of $\mathcal{M}$ contain a and b, respectively. Notice that M is an α-neighboring cell of a vertex of G of degree at least three, and thus ∂M appears on the drawing of H by the construction of H. Therefore, ∂A_t intersects ∂M only at vertices of H, say v. Note that $v \in \mathsf{bd}(t)$, and a and b are contained in cells of $\mathcal{M}_{\mathsf{nb}}(v)$.

Now consider the case that M_a and M_b contain vertices of degree two only. Thus a and b are degree-2 vertices of G. Consider a maximal chain π of degree-2 vertices of G containing ab. It is a part of the drawing of H, and thus ∂A_t intersects π only at vertices of π. Therefore, ab is intersected by ∂A_t only when a or b lies on ∂A_t. Without loss of generality, assume that a lies on ∂A_t. This means that $a \in \mathsf{bd}(t)$, and both a and b are contained in cells of $\mathcal{M}_{\mathsf{nb}}(a)$ in this case. Thus in any case, the claim holds. ◀

4 Properties of Vertex-Disjoint Cycles

For a unit disk graph G drawn in the plane with a map $\mathcal{M}$, we analyze properties of a set of k vertex-disjoint cycles in G. As a preprocessing, we remove all vertices not contained in any cycle of G. In particular, we recursively remove a vertex of degree at most one from G. Note that the resulting graph is also a unit disk graph. Then we show that if G has more than ck vertices of degree at least three for a constant c, (G, k) is a **yes**-instance. In this case, we immediately return the answer. Otherwise, G has k vertex-disjoint cycles with the bounded packedness property and the quasi-planar property defined as follows.

A set Γ of vertex-disjoint cycles of G is *c-packed* if at most c vertices contained in a cell of $\mathcal{M}$ lie on the cycles of Γ', and every cycle of $\Gamma \setminus \Gamma'$ is a triangle, where Γ' is the set of cycles of Γ visiting at least vertices from different cells of $\mathcal{M}$. The set Γ is *quasi-planar* if each cycle of Γ is not self-crossing, and for any set Π' of subsets of non-triangle cycles of Γ, the intersection graph of Π' is $K_{z,z}$-free for a constant z depending only on the maximum number of paths of Π' containing a common edge. Two paths of Π' might come from the same cycle of Γ, and they can share a common edge. Here, the intersection graph is defined as the graph where a vertex corresponds to a non-triangle cycle of Γ, and two vertices are connected by an edge if and only if their corresponding cycles cross in their drawing.



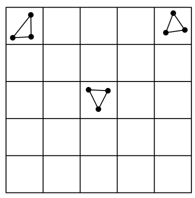


Figure 4 The α-neighboring cells around M. If there are three vertices v_1, v_2 and v_3 in M such that the neighboring vertices of them on Γ' are contained in M_1 and M_2, the three cycles containing v_1, v_2 and v_3, respectively, can be replaced with the three triangles.

▶ **Lemma 3.** *Suppose G has more than ck vertices of degree at least three after the preprocessing step for a constant c. Then (G, k) is a **yes**-instance. Otherwise, G has $O(1)$-packed and quasi-planar k vertex-disjoint cycles if and only if G has k vertex-disjoint cycles.*

Sketch of the proof. For the first part, we consider the case that all vertices of G has degree at least three in this sketch. If two edges xy and $x'y'$ of G cross, then three of four vertices, say x, y and x' form a triangle. Using this, we remove three vertices from the pair of crossing edges until the remaining graph becomes planar. If the removal happens k times, (G, k) is a **yes**-instance. Otherwise, we consider the (planar) dual graph of the remaining graph H. The dual graph has $c'k$ vertices by the Euler's formula and by the construction. Since the dual graph is 5-colorable, it contains $c'k/5$ independent vertices, and they correspond to $c'k/5$ faces of H. Note that the boundary cycles of such faces are vertex-disjoint cycles in G. By setting c so that $c'k/5 \geq k$, the claim holds in the case that all vertices have degree at least three. We can prove the claim for the general case similarly by contracting the edges whose both endpoints are degree-2 vertices.

For the second part, let Γ be a set of k vertex-disjoint cycles of G that minimizes the number of cycles of Γ visiting at least two vertices from different cells. Let Γ' be the set of these cycles. Let M be a cell of $\mathcal{M}$ and Q be the set of vertices of $V \cap M$ lying on the cycles of Γ'. Suppose $|Q| > 3\beta^2$, where β is the maximum number of α-neighboring cells of M. Since all neighbors of the vertices of Q are contained in β cells, there is a pair (M_1, M_2) of such cells and three cycles of Γ' such that each cycle contains an edge connecting the vertices of M and M_1, and an edge connecting the vertices of M and M_2. Then we can replace three cycles into three triangles consisting of the vertices lying on three cycles contained in M, M_1 and M_2, respectively. This contradicts the choice of Γ. See Figure 4.

For the last part, we let Γ be a maximum-sized set of vertex-disjoint cycles of G that is $O(1)$-packed. If it is not unique, we choose the one that minimizes the total number of edges of Γ among them. Consider the intersection graph of Π, where Π is a set of subpaths of non-triangle cycles of Γ. Since it is $O(1)$-packed, the number of crossing points of paths of Π contained in the same cell is $O(1)$. If the intersection graph contains $K_{z,z}$ as a subgraph for a sufficiently large constant z, we can always find four paths π_1, π_2, π'_1 and π'_2 of Π such that the pairwise Euclidean distances between $c_{1,1}, c_{1,2}, c_{2,1}$ and $c_{2,2}$ are at least five, where $c_{i,j}$ is a crossing point between π_i and π'_j. This means that the endpoints of the edges involved in such crossings are all distinct. For each pair $(xy, x'y')$ of crossing edges, we can show that three of $\{x, y, x', y'\}$ form a triangle in G. Then we replace the cycles of Γ containing π_1, π_2, π'_1 and π'_2 into four vertex-disjoint triangles. In this way, the number of edges participating in Γ decreases while the number of cycles of Γ remains the same, or the number of cycles of Γ increases. This violates the choice of Γ. ◀

5 Generalization of Catalan Bounds to Crossing Circular Arcs

It is well-known that, for a fixed set P of n points on a circle, the number of different sets of pairwise non-crossing circular arcs having their endpoints on P is the n-th Catalan number, which is $2^{O(n)}$. Here, two circular arcs are *non-crossing* if they are disjoint or one circular arc contains the other circular arc. This fact is one of main tools used in the ETH-tight algorithm for the planar cycle packing problem: Given a noose γ of a planar graph G visiting m vertices of G, the parts of the cycles of Γ contained in the interior of γ corresponds to a set of pairwise non-crossing circular arcs having their endpoints on fixed m points, where Γ denotes a set of vertex-disjoint cycles of G crossing γ. Although two vertex-disjoint cycles in unit disk graphs can cross in their drawing, there exists a set of k vertex-disjoint cycles of G with the quasi-planar property due to Lemma 3. To make use of this property, we generalize the Catalan bound on non-crossing circular arcs to circular arcs which can cross.

The *circular arc crossing graph*, CAC graph in short, is a variation of the circular arc graph. For a fixed set P of points on a unit circle, consider a set $\mathcal{C}$ of circular arcs on the unit circle connecting two points of P such that no arcs share their endpoints. The CAC graph of $\mathcal{C}$ is defined as the graph whose vertices correspond to circular arcs of $\mathcal{C}$, and two vertices are connected by an edge if and only if their corresponding circular arcs are crossing. Here, we say two arcs on the unit circle are *crossing* if they are intersect and none of them contains the other arc. Note that the CAC graph G_{cac} of $\mathcal{C}$ depends only on the pairing $\mathcal{P}$ of the endpoints of the arcs of $\mathcal{C}$. If we do not want to specify circular arcs, we simply refer to G_{cac} as the CAC graph of $\mathcal{P}$. With a slight abuse of term, we say a set $\mathcal{C}$ of circular arcs (or its pairing) is $K_{z,z}$-free if its CAC graph is $K_{z,z}$-free. Note that a set of pairwise non-crossing circular arcs is $K_{1,1}$-free since its CAC graph is edgeless.

▶ **Lemma 4.** *The number of $K_{z,z}$-free sets of circular arcs over a fixed set P is $2^{O_z(|P|)}$.*

Sketch of the proof. For the convenience, we let $P = \{1, 2, \dots, 2m\}$ be the cyclic sequence of the elements of $\mathbb{Z}_{2m}$. For a $K_{z,z}$-free set $\mathcal{C}$ of circular arcs, an arc of $\mathcal{C}$ with counterclockwise endpoint a and clockwise endpoint b for $\mathbb{Z}_{2m}$ is denoted by $[a, b]$.

We prove a weaker claim that the number of $K_{z,z}$-free sets $\mathcal{C}$ such that no arc of $\mathcal{C}$ contains any other arc of $\mathcal{C}$ is $2^{O_z(|P|)}$. In this case, for a circular arc $[a, b]$ in $\mathcal{C}$, at least $\frac{b-a}{2}$ circular arcs of $\mathcal{C}$ contain a, and these arcs induce a clique of size $\frac{b-a}{2}$ in the CAC graph of $\mathcal{C}$. Since this graph is K_{2z}-free, we have $b - a \leq 4z$. Thus, for each a, there are $4z$ different candidates for the other endpoint b of an arc $[a, b]$. Thus, the number of different $\mathcal{C}$ is $(4z)^{|P|} = 2^{O_z(|P|)}$.

Using this observation, we prove the claim that the number of all $K_{z,z}$-free sets $\mathcal{C}$ is $2^{O_z(|P|)}$. We decompose the arcs of $\mathcal{C}$ into several *layers* as follows. In each iteration, we choose all arcs of $\mathcal{C}$ not contained in any other arcs of $\mathcal{C}$, and remove them from $\mathcal{C}$. The *level* of γ is defined as the index of the iteration when γ is removed. Then the followin claim holds.

▷ Claim 5. An arc of level i contains at most z endpoints of the arcs of level less than $i - z$.

For each integer $i \leq m$, we denote the set of endpoints of all circular arcs of level at most i by P_i. Then we have $P_1 \subset P_2 \subset \ldots \subset P_m = P$. For a subset P' of P, we define the *partial order* $\chi_{P'} : P' \to [1, |P'|]$ such that $\chi_{P'}(a) = b$ if a is the b-th endpoint of P' (starting from 1). We simply write $\chi_{i,j} = \chi_{P_j \setminus P_{i-1}}$ for every $i \leq j$. Given a partial order $\chi_{i,j}$, consider the mapping that maps each circular arc $[a, b]$ into $[\chi_{i,j}(a), \chi_{i,j}(b)]$, where $a, b \in P_j \setminus P_{i-1}$. We define the *quotient* $Q_{i,j}$ as the set of circular arcs $[\chi_{i,j}(a), \chi_{i,j}(b)]$ on the ground set $[1, 2, \dots |P_j \setminus P_{i-1}|]$. The CAC graph of $Q_{i,j}$ is isomorphic to the subgraph of the CAC graph of $\mathcal{C}$, and therefore it is $K_{z,z}$-free.

Then we analyze the number of $K_{z,z}$-free sets of circular arcs over P. For this, we first show that the number of different quotients $Q_{i-z,i}$ is $2^{O(|P_i \setminus P_{i-z}|)}$ similarly to the simplest case that we mentioned above. Then we compute the number of different quotients $Q_{1,i}$ under a fixed $Q_{1,i-1}$. Now it equals the number of ways to fix the position of P_{i-z} on $\chi_{1,i}$. Let a, b be the consecutive endpoints of $P_i \setminus P_{i-z}$. If both a and b are contained in P_{i-1}, the number of endpoints of P_{i-z} contained in $[a,b]$ is fixed under $Q_{1,i-1}$. On the other hand, if a is an endpoint of an arc of level i, there is an arc of level $i-1$ containing $[a,b]$. Then the number of endpoints of P_{i-z-1} contained in $[a,b]$ is $O(1)$ due to Claim 5. Using this observation, we can show that the number of different quotients $Q_{1,i}$ is $2^{O(|P_i \setminus P_{i-z-1}|)}$. Then the number of different $K_{z,z}$-free sets of circular arcs is

$$\exp(\sum_i O(|P_i \setminus P_{i-z-1}|)) = \exp(O_z(|P|)). \qquad ◀$$

Moreover, we can compute all $K_{z,z}$-free sets of circular arcs over P in $2^{O_z(|P|)}$ time. See the full version for details.

6 Dynamic Programming on an Surface Cut Decomposition

In this section, we are given a unit disk graph $G = (V, E)$ drawn in the plane along with a map $\mathcal{M}$ and an sc-decomposition $(T, \mathcal{A})$ of clique-weighted width ω. Then we present a $2^{O(\omega)} n^{O(1)}$-time dynamic programming algorithm that computes a maximum number of vertex-disjoint cycles of G. Throughout this section, let Γ be a maximum-sized set of vertex-disjoint cycles which is quasi-planar and $O(1)$-packed. Let $\mathcal{M}_{\mathsf{f}}$ be the finer subdivision of $\mathcal{M}$ formed by subdividing each cell of $\mathcal{M}$ into *finer cells* along ∂A for all pieces A corresponding to the leaf nodes of T. We call a cycle of Γ an *intra-cell* cycle if its vertices are contained in a single finer cell of $\mathcal{M}_{\mathsf{f}}$, and an *inter-cell* cycle, otherwise. Notice that an intra-cell cycle is a triangle since Γ is $O(1)$-packed. To make the description easier, we sometimes consider a set of vertex-disjoint cycles and paths of G as a *graph* consisting of disjoint paths (path components) and cycles (cycle components) if it is clear from the context.

For a set Γ' of paths and cycles of G, we let $\mathsf{end}(\Gamma')$ be the set of end vertices of the *paths* of Γ', and $V(\Gamma')$ be the set of all vertices of *paths and cycles* of Γ'. Let $\mathcal{M}_t$ be the set of finer cells of $\mathcal{M}_{\mathsf{f}}$ having vertices of $V(\mathsf{cut}(t)) \cap A_t$. The clique-weight of the finer cells in $\mathcal{M}_t$ is $O(\omega)$ as for each cell M of $\mathcal{M}$, $V(G) \cap M$ is decomposed into $O(1)$ subsets $V(G) \cap M_{\mathsf{f}}$ for finer cells M_{f} of $\mathcal{M}_{\mathsf{f}}$ by **(C4)**.

The interaction between Γ and $\mathsf{cut}(t)$ can be characterized as $(\Lambda, V(\Delta), \mathcal{P})$ where Λ be the subgraph of Γ induced by $\mathsf{cut}(t)$, and Δ be the set of intra-cell cycles of Γ contained in a cell of $\mathcal{M}_t$. Moreover, $\mathcal{P}$ is the pairing of the vertices of $\mathsf{end}(\Lambda)$ contained in A_t such that the vertices in the same pair belong to the same component of the subgraph of Γ induced by $V \cap A_t$. We call $(\Lambda, V(\Delta), \mathcal{P})$ the *t-signature* of Γ.

Since we are not given Γ in advance, we try all possible tuples which can be the signatures of Γ, and find an optimal solution for each of such signatures. We define the subproblem for each *valid* tuple $(\widetilde{\Lambda}, \widetilde{V}, \widetilde{\mathcal{P}})$. We say a tuple $(\widetilde{\Lambda}, \widetilde{V}, \widetilde{\mathcal{P}})$ is *valid* if for every finer cell M of $\mathcal{M}_t$

- $\widetilde{\Lambda}$ induces vertex-disjoint paths and cycles consisting of edges of $\mathsf{cut}(t)$,
- $\widetilde{V}$ is a subset of $\{v \in M' \mid M' \in \mathcal{M}_t\}$ with $\widetilde{V} \cap V(\widetilde{\Lambda}) = \emptyset$ and $|M \cap (V \setminus \widetilde{V})| = O(1)$, and
- $\widetilde{\mathcal{P}}$ is a pairing of all vertices of $\mathsf{end}(\widetilde{\Lambda})$ contained in A_t.

Notice that the t-signature of Γ is also a valid tuple. For each valid tuple Q for a node t, we compute the maximum number of vertex-disjoint cycles of Γ' contained in G_t over all sets Γ' of vertex-disjoint cycles and paths of $G[E(G_t) \cup \mathsf{cut}(t)]$ whose t-signature is Q.[5]

Deep analysis of crossing patterns. We give more details of the methods presented in Section 2.2 as follows. In order to reduce the number of valid tuples, we analyze the crossing pattern of inter-cell cycles of Γ more carefully. Let t be a node of T, and let G_t be the subgraph of G induced by $V \cap A_t$. Let Π be the set of path components of the subgraph of Γ induced by $V \cap A_t$. Notice that a path of Π is a part of an inter-cell cycle of Γ by the condition **(C3)**. For a path π of Π, its endpoint is incident to a cut edge of A_t in Γ. The path obtained from π by adding the two cut edges at the end of π is denoted by $\overleftrightarrow{\pi}$. In the following, we define the ordering of paths π of Π with respect to a crossing point between $\overleftrightarrow{\pi}$ and ∂A_t. For this purpose, let $\bar{\pi}$ be the path (polygonal curve) obtained from $\overleftrightarrow{\pi}$ by removing parts of its end edges maximally such that the endpoints of $\bar{\pi}$ lie on ∂A_t. We call $\bar{\pi}$ the *anchored path* of π. Let $\bar{\Pi}$ be the set of the anchored paths of the paths of Π. A path of $\bar{\Pi}$ is fully contained in A_t by the definition of $\mathsf{cut}(t)$.

Simple case: ∂A_t is a single curve. Given a fixed set P of $O(\omega)$ vertices on ∂A_t (indeed, $\mathsf{end}(\bar{\Pi})$), our goal is to compute a small number of pairings of P one of which is a correct pairing of $\bar{\Pi}$. We first focus on the simple case that all points of $\mathsf{end}(\bar{\Pi})$ lie on a single boundary curve C of A_t. We compute $2^{O(\omega)}$ pairings one of which is a correct pairing as follows. For a path $\bar{\pi}$ of $\bar{\Pi}$, we represent it as the pair of its end vertices on C. Then the circular arc crossing graph of all pairs of $\bar{\Pi}$ is isomorphic to a subgraph of the intersection graph of Π, and it is $K_{z,z}$-free by the quasi-planar property. That is, for any two paths $\bar{\pi}$ and $\bar{\pi}'$ of $\bar{\Pi}$ ending at (a, b) and (a', b'), respectively and if a, a', b and b' appear along C in this order, the two paths must cross as they cannot cross ∂A_t. Using this, we simply enumerate all pairings of P whose corresponding circular arc graphs are $K_{z,z}$-free by Lemma 4. The number of such pairings is $2^{O(|P|)} = 2^{O(\omega)}$, and we can compute them in $2^{O(\omega)}$ time.

General case: ∂A_t has more than one curves. Now we consider the general case where two endpoints of $\bar{\pi}$ are contained in different boundary curves of ∂A_t. In particular, let (C, C') be a pair of boundary curves of A_t. To handle the paths $\bar{\pi}_{x,y}$ of $\bar{\Pi}$ with $x \in C$ and $y \in C'$, we connect C and C' with a curve $\lambda = \lambda(C, C')$. This curve intersects C and C' only at its endpoints and does not contain any endpoints of the paths of $\bar{\Pi}$. Subsequently, we obtain the closed curve $\bar{\lambda}$ consisting of C, C' and λ such that λ appears in $\bar{\lambda}$ twice. Since none of the endpoints of a path in $\bar{\Pi}$ lie on λ, we can orderly arrange the endpoints of paths in $\bar{\Pi}$ so that they lie on $C \cup C'$. This arrangement follows along $\bar{\lambda}$.

We show that for two paths $\bar{\pi}_{x,y}$ and $\bar{\pi}_{x',y'}$ with $x, x' \in C$ and $y, y' \in C'$, if they are cross-ordered, and their *crossing numbers* are the same in modulo 2, then $\bar{\pi}_{x,y}$ and $\bar{\pi}_{x',y'}$ cross. Here, the crossing number of a path $\bar{\pi}_{x,y}$ is defined as the number of times that $\bar{\pi}_{x,y}$ crosses $\lambda(C, C')$. Note that $\bar{\pi}_{x,y}$ and $\bar{\pi}_{x',y'}$ never cross either C or C'. See also Figure 5.

▶ **Lemma 6.** *For any two paths $\bar{\pi}_{x,y}$ and $\bar{\pi}_{x',y'}$ with $x, x' \in C$ and $y, y' \in C'$ for two boundary curves C and C' of ∂A_t, if they are cross-ordered, and their crossing numbers are the same in modulo 2, then $\bar{\pi}_{x,y}$ and $\bar{\pi}_{x',y'}$ cross.*

[5] The signature is defined for a set of disjoint cycles of G, but it can be extended to a set of disjoint paths and cycles of $G[E(G_t) \cup \mathsf{cut}(t)]$ in a straightforward manner.

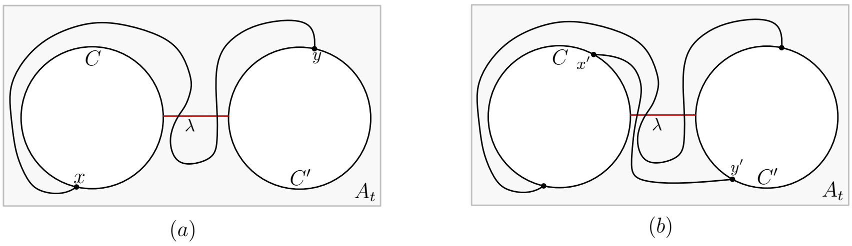


Figure 5 (a) The union of C, C' and λ (red curve) forms a closed curve. The path $\bar{\pi}_{x,y}$ has crossing number 2. (b) Illustrates two paths whose four endpoints are cross-ordered. Their crossing numbers differ in modulo 2.

Imagine that we subdivide $\bar{\Pi}$ into $O(1)$ subsets such that all paths in the same subset have their endpoints on the same boundary curves of A, and their crossing numbers are the same. Then the circular arc crossing graph of each subset is $K_{z,z}$-free, and thus the number of such pairings for $\bar{\Pi}$ is $2^{O(\omega)}n^{O(1)}$. To use this, we design an improved dynamic programming algorithm in the full version of this paper, which runs in time $2^{O(\omega)}n^{O(1)}$.

▶ **Theorem 7.** *Given a unit disk graph G with its geometric representation and an integer k, Cycle Packing can be solved in $2^{O(\sqrt{k})}n^{O(1)}$ time.*

Proof. We compute a map sparsifier of G on a grid map $\mathcal{M}$. Then we repeatedly remove a vertex of degree at most one from G. By Lemma 3, if G has $\omega(k)$ vertices of degree at least three, (G, k) is a **yes**-instance and we can compute a solution in polynomial time. Otherwise, we compute an sc-decomposition of clique-weighted width $O(\sqrt{k})$ by Theorem 2. We can solve the problem in $2^{O(\sqrt{k})}n^{O(1)}$ time using a dynamic programming running in time single exponential in the width of an sc-decomposition. ◀

References

1. Shinwoo An and Eunjin Oh. Feedback Vertex Set on Geometric Intersection Graphs. In *Proceedings of the 32nd International Symposium on Algorithms and Computation (ISAAC 2021)*, pages 47:1–47:12, 2021.
2. Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. True contraction decomposition and almost eth-tight bipartization for unit-disk graphs. In *38th International Symposium on Computational Geometry (SoCG 2022)*, pages 11:1–11:16, 2022.
3. Sujoy Bhore, Paz Carmi, Sudeshna Kolay, and Meirav Zehavi. Parameterized study of steiner tree on unit disk graphs. *Algorithmica*, 85:133–152, 2023.
4. Hans L Bodlaender, Eelko Penninkx, and Richard B Tan. A linear kernel for the k-disjoint cycle problem on planar graphs. In *Proceedings of the 19th International Symposium on Algorithms and Computation (ISAAC 2008)*, pages 306–317, 2008.
5. Sergio Cabello and Miha Jejčič. Shortest paths in intersection graphs of unit disks. *Computational Geometry*, 48(4):360–367, 2015.
6. Timothy M Chan. Finding triangles and other small subgraphs in geometric intersection graphs. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2023)*, pages 1777–1805, 2023.
7. Timothy M Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *Journal of Computational Geometry*, 10(2):3–20, 2019.
8. Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Joham MM van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 150–159, 2011.

9 Mark de Berg, Hans L Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C van der Zanden. A framework for Exponential-Time-Hypothesis–tight algorithms and lower bounds in geometric intersection graphs. *SIAM Journal on Computing*, 49(6):1291–1331, 2020.

10 Hristo N Djidjev. Weighted graph separators and their applications. In *Proceedings of the 5th Annual European Symposium on Algorithms (ESA 1997)*, pages 130–143, 1997.

11 Frederic Dorn, Fedor V Fomin, and Dimitrios M Thilikos. Catalan structures and dynamic programming in H-minor-free graphs. *Journal of Computer and System Sciences*, 78(5):1606–1622, 2012.

12 Frederic Dorn, Eelko Penninkx, Hans L Bodlaender, and Fedor V Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. *Algorithmica*, 58(3):790–810, 2010.

13 Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

14 Louis Esperet, Sébastien Julliot, and Arnaud de Mesmay. Distributed coloring and the local structure of unit-disk graphs. *Theoretical Computer Science*, 944:113674, 2023.

15 Fedor V Fomin, Petr A Golovach, Tanmay Inamdar, and Saket Saurabh. ETH tight algorithms for geometric intersection graphs: Now in polynomial space. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*, pages 21:1–21:16, 2021.

16 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Finding, hitting and packing cycles in subexponential time on unit disk graphs. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, pages 65:1–65:15, 2017.

17 Fedor V Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Decomposition of map graphs with applications. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, pages 60:1–60:15, 2019.

18 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. ETH-tight algorithms for long path and cycle on unit disk graphs. In *Proceedings of the 36th International Symposium on Computational Geometry (SoCG 2020)*, pages 44:1–44:18, 2020.

19 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 1563–1575, 2012.

20 Zachary Friggstad and Mohammad R Salavatipour. Approximability of packing disjoint cycles. *Algorithmica*, 60(2):395–400, 2011.

21 Sariel Har-Peled and Amir Nayyeri. A simple algorithm for computing a cycle separator. *CoRR*, abs/1709.08122, 2017.

22 Refael Hassin and Shlomi Rubinstein. An approximation algorithm for maximum triangle packing. *Discrete applied mathematics*, 154(6):971–979, 2006.

23 Weizhi Hong, Yingli Ran, and Zhao Zhang. Parallel algorithms for minimum general partial dominating set and maximum budgeted dominating set in unit disk graph. *Theoretical Computer Science*, 932:13–20, 2022.

24 Sangram K Jena and Gautam K Das. Vertex-edge domination in unit disk graphs. *Discrete Applied Mathematics*, 319:351–361, 2022.

25 Haim Kaplan, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth. Routing in unit disk graphs. *Algorithmica*, 80(3):830–848, 2018.

26 Ken-ichi Kawarabayashi and Bruce Reed. Odd cycle packing. In *Proceedings of the forty-second ACM symposium on Theory of computing (STOC 2010)*, pages 695–704, 2010.

27 Philip N Klein, Shay Mozes, and Christian Sommer. Structured recursive separator decompositions for planar graphs in linear time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing (STOC 2013)*, pages 505–514, 2013.

28 Michael Krivelevich, Zeev Nutov, Mohammad R Salavatipour, Jacques Verstraete, and Raphael Yuster. Approximation algorithms and hardness results for cycle packing problems. *ACM Transactions on Algorithms*, 3(4):48:1–48:21, 2007.

29 Daniel Lokshtanov, Amer E Mouawad, Saket Saurabh, and Meirav Zehavi. Packing cycles faster than Erdös–Pósa. *SIAM Journal on Discrete Mathematics*, 33(3):1194–1215, 2019.

30 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. Subexponential parameterized algorithms on disk graphs. In *33rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*, pages 2005–2031, 2022.

31 Dániel Marx. Parameterized complexity of independence and domination on geometric graphs. In *Proceedings of the 2nd International Conference on Parameterized and Exact Computation (IWPEC 2006)*, pages 154–165, 2006.