




Finding, Hitting and Packing Cycles in Subexponential Time on Unit Disk Graphs

Fedor V. Fomin¹ · Daniel Lokshtanov^{1,2} · Fahad Panolan¹  · Saket Saurabh^{1,3} · Meirav Zehavi⁴

Received: 11 September 2017 / Revised: 13 November 2018 / Accepted: 13 December 2018 /
Published online: 7 January 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

We give algorithms with running time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ for the following problems. Given an n -vertex unit disk graph G and an integer k , decide whether G contains

- a path on exactly/at least k vertices,
- a cycle on exactly k vertices,
- a cycle on at least k vertices,
- a feedback vertex set of size at most k , and
- a set of k pairwise vertex-disjoint cycles.

For the first three problems, no subexponential time parameterized algorithms were previously known. For the remaining two problems, our algorithms significantly outperform the previously best known parameterized algorithms that run in time $2^{\mathcal{O}(k^{0.75} \log k)} \cdot n^{\mathcal{O}(1)}$. Our algorithms are based on a new kind of tree decompositions of unit disk graphs where the separators can have size up to $k^{\mathcal{O}(1)}$ and there exists a solution that crosses every separator at most $\mathcal{O}(\sqrt{k})$ times. The running times of our algorithms are optimal up to the $\log k$ factor in the exponent, assuming the exponential time hypothesis.

Keywords Longest path · Longest cycle · Cycle packing · Feedback vertex set · Unit disk graph · Unit square graph · Parameterized complexity

Mathematics Subject Classification 68W01 · 68W40 · 68Q25

Editor in Charge: Kenneth Clarkson

The preliminary version of the paper appeared in the proceedings of ICALP 2017. The research leading to the results of the paper is supported by Pareto-Optimal Parameterized Algorithms, ERC Starting Grant 715744, Parameterized Approximation, ERC Starting Grant 306992, Rigorous Theory of Preprocessing, ERC Advanced Investigator Grant 267959, and NFR MULTIVAL project.

Extended author information available on the last page of the article

1 Introduction

Unit disk graphs are the intersection graphs of unit circles in the plane. That is, given n -unit circles in the plane, we have a graph G where each vertex corresponds to a circle such that there is an edge between two vertices when the corresponding circles intersect. Unit disk graphs form one of the most well studied graph classes in computational geometry because of their use in modelling optimal facility location [41] and broadcast networks such as wireless, ad hoc and sensor networks [25,34,43]. These applications have led to an extensive study of NP-complete problems on unit disk graphs in the realms of computational complexity and approximation algorithms. We refer the reader to [11,19,30] and the citations therein for these studies. However, these problems remain hitherto unexplored in the light of parameterized complexity with exceptions that are few and far between [1,9,23,33,39].

In this paper we consider the following basic problems about finding, hitting and packing cycles on unit disk graphs from the viewpoint of parameterized algorithms. For a given graph G and integer k ,

- EXACT k -CYCLE asks whether G contains a cycle on exactly k vertices,
- LONGEST CYCLE asks whether G contains a cycle on at least k vertices,
- FEEDBACK VERTEX SET asks whether G contains a vertex set S of size k such that the graph $G \setminus S$ is acyclic, and
- CYCLE PACKING asks whether G contains a set of k pairwise vertex-disjoint cycles.

Along the way, we also study LONGEST PATH (decide whether G contains a path on exactly/at least k vertices) and SUBGRAPH ISOMORPHISM (SI). In SI, given *connected* graphs G and H on n and k vertices, respectively, the goal is to decide whether there exists a subgraph in G that is isomorphic to H . Throughout the paper we assume that a unit disk graph is given by a set of n points in the Euclidean plane and there is a graph where vertices correspond to these points and there is an edge between two vertices if and only if the Euclidean distance between the two points is at most 2.

In parameterized complexity each of these problems serves as a testbed for development of fundamental algorithmic techniques such as color-coding [2], the polynomial method [4,35,36,42], matroid based techniques [21] for LONGEST PATH and LONGEST CYCLE, and kernelization techniques for FEEDBACK VERTEX SET [40]. We refer to [13] for an extensive overview of the literature on parameterized algorithms for these problems. For example, the fastest known algorithms solving LONGEST PATH are the $1.66^k \cdot n^{\mathcal{O}(1)}$ time randomized algorithm of Björklund et al. [4], and the $2.597^k \cdot n^{\mathcal{O}(1)}$ time deterministic algorithm of Zehavi [44]. Moreover, unless the exponential time hypothesis (ETH) of Impagliazzo et al. [31] fails, none of the problems above can be solved in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ [31].

While all these problems remain NP-complete on planar graphs, substantially faster—*subexponential*—parameterized algorithms are known on planar graphs. In particular, by combining the bidimensionality theory of Demaine et al. [14] with efficient algorithms on graphs of bounded treewidth [18], LONGEST PATH, LONGEST CYCLE, FEEDBACK VERTEX SET and CYCLE PACKING are solvable in time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ on planar graphs. The parameterized subexponential “tractability” of such problems can be extended to graphs excluding some fixed graph

as a minor [16]. The bidimensionality arguments cannot be applied to EXACT k -CYCLE and this was one of the motivations for developing the new pattern-covering technique, which is used to give a randomized algorithm for EXACT k -CYCLE running in time $2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)}$ on planar and apex-minor-free graphs [20]. The bidimensionality theory was also used to design (efficient) polynomial time approximation scheme ((E)PTAS) [15,22] and polynomial kernelization [24] on planar graphs.

It would be interesting to find generic properties of problems, similar to the theory of bidimensionality for planar-graph problems, that could guarantee the existence of subexponential parameterized algorithms or (E)PTAS on geometric classes of graphs, such as unit disk graphs. The theory of (E)PTAS on geometric classes of graphs is extremely well developed and several methods have been devised for this purpose. This includes methods such as shifting techniques, geometric sampling and bidimensionality theory [12,23,26–28,30,38]. However, we are still very far from a satisfactory understanding of the “subexponential” phenomena for problems on geometric graphs. We know that some problems such as INDEPENDENT SET and DOMINATING SET, which are solvable in time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ on planar graphs, are W[1]-hard on unit disk graphs and thus the existence of an algorithm of running time $f(k) \cdot n^{\mathcal{O}(1)}$ is highly unlikely for any function f [37]. The existence of a vertex-linear kernel for some problems on unit disk graphs such as VERTEX COVER [10] or CONNECTED VERTEX COVER [33] combined with an appropriate separation theorem [1,9,39] yields a parameterized subexponential algorithm. A subset of the authors of this paper used a different approach based on bidimensionality theory to obtain subexponential algorithms of running time $2^{\mathcal{O}(k^{0.75} \log k)} \cdot n^{\mathcal{O}(1)}$ on unit disk graphs for FEEDBACK VERTEX SET and CYCLE PACKING in [23]. No parameterized subexponential algorithms on unit disk graphs for LONGEST PATH, LONGEST CYCLE, and EXACT k -CYCLE were known prior to our work.

Our results. We design subexponential parameterized algorithms, with running time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$, for EXACT k -CYCLE, LONGEST CYCLE, LONGEST PATH, FEEDBACK VERTEX SET and CYCLE PACKING on unit disk graphs and unit square graphs. It is also possible to show by known NP-hardness reductions for problems on unit disk graphs [11] that an algorithm of running time $2^{o(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ for any of our problems on unit disk graphs would imply that ETH fails. Hence our algorithms are asymptotically almost tight. Along the way we also design Turing kernels (in fact, many-to-one) for EXACT k -CYCLE, LONGEST CYCLE, LONGEST PATH and SI. That is, we give a polynomial time algorithm that given an instance of EXACT k -CYCLE or LONGEST CYCLE or LONGEST PATH or SI, produces polynomially many reduced instances of size polynomial in k such that the input instance is a YES-instance if and only if one of the reduced instances is a YES-instance. As a byproduct of this we obtain a $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ time algorithm for SI when G is a unit disk graph and H is an arbitrary connected graph. It is noteworthy to remark that a simple disjoint union trick implies that EXACT k -CYCLE, LONGEST CYCLE, LONGEST PATH, and SI do not admit a polynomial kernel on unit disk graphs [7]. Finally, we remark that we do not use Turing kernels to design our subexponential time algorithms except for EXACT k -CYCLE. The subexponential time

parameterized algorithm for EXACT k -CYCLE also uses a “double layering” of Baker’s technique [3].

All our subexponential time algorithms have the following theme in common. If an input n -vertex unit disk graph G contains a clique of size $\text{poly}(k)$ (such a clique can be found in polynomial time), then we have a trivial YES-instance or NO-instance, depending on the problem. Otherwise, we show that the unit disk graph G in a YES-instance of the problem admits, sometimes after a polynomial time preprocessing, a specific type of (ω, Δ, τ) -decomposition, where the meaning of ω , Δ and τ is as follows. The vertex set of G is partitioned into cliques C_1, \dots, C_d , each of size at most $\omega = k^{\mathcal{O}(1)}$. We also require that after contracting each of the cliques C_i to a single vertex, the maximum vertex degree Δ of the obtained graph \tilde{G} is $\mathcal{O}(1)$, while the treewidth τ of \tilde{G} is $\mathcal{O}(\sqrt{k})$. Moreover, the corresponding tree decomposition of \tilde{G} can be constructed efficiently. We use the tree decomposition of \tilde{G} to construct a tree decomposition of G by “uncontracting” each of the contracted cliques C_i . While the width of the obtained tree decomposition of G can be of order $\omega \cdot \tau = k^{\mathcal{O}(1)}$, we show that each of our parameterized problems can be solved in time $f(\Delta) \cdot \omega^{f(\Delta) \cdot \tau}$. Here we use dynamic programming over the constructed tree decomposition of G , however there is a twist from the usual way of designing such algorithms. This part of the algorithm is problem-specific—in order to obtain the claimed running time, we have to establish a very specific property for each of the problems. Roughly speaking, the desired property of a problem is that it always admits an optimal solution such that for every pair of adjacent bags X, Y of the tree decomposition of G , the number of edges of this solution “crossing” a cut between X and Y is $\mathcal{O}(\sqrt{k})$. We remark that the above decomposition is *only* given in the introduction to present our ideas for all the algorithms in a unified way and the term (ω, Δ, τ) -decomposition is not defined in the technical sections.

2 Preliminaries

For a positive integer t , we use $[t]$ as a shorthand for $\{1, 2, \dots, t\}$. Given a function $f: A \rightarrow B$ and a subset $A' \subseteq A$, let $f|_{A'}$ denote the restriction of the function f to the domain A' . For a function $f: A \rightarrow B$ and $B' \subseteq B$, $f^{-1}(B')$ denotes the set $\{a \in A : f(a) \in B'\}$. For $t, t' \in \mathbb{N}$, a set $[t] \times [t']$, $i \in [t]$ and $j \in [t']$ we use $(*, j)$ and $(i, *)$ to denote the sets $\{(i', j) : i' \in [t]\}$ and $\{(i, j') : j' \in [t']\}$, respectively. For a set U , we use 2^U to denote the power set of U .

Graph Theory. We use standard notation and terminology from the book of Diestel [17] for graph-related terms which are not explicitly defined here. Given a graph G , $V(G)$ and $E(G)$ denote its vertex-set and edge-set, respectively. When the graph G is clear from context, we denote $n = |V(G)|$ and $m = |E(G)|$. Given $U \subseteq V(G)$, we let $G[U]$ denote the subgraph of G induced by U , and we let $G \setminus U$ denote the graph $G[V(G) \setminus U]$. For an edge subset E , we use $V(E)$ to denote the set of endpoints of edges in E and $G[E]$ to denote the graph $(V(E), E)$. For $X, Y \subseteq V(G)$, we use $E(X)$ and $E(X, Y)$ to denote the edge sets $\{\{u, v\} \in E(G) : u, v \in X\}$ and $\{\{u, v\} \in E(G) : u \in X, v \in Y\}$, respectively. Moreover, we let $N(U)$ denote

the open neighborhood of G . In case $U = \{v\}$, we denote $N(v) = N(U)$. Given an edge $e = \{u, v\} \in E(G)$, we use G/e to denote the graph obtained from G by contracting the edge e . In other words, G/e denotes the graph on the vertex-set $(V(G) \setminus \{u, v\}) \cup \{x_{\{u,v\}}\}$, where $x_{\{u,v\}}$ is a new vertex, and the edge-set $E(G/e) = E(G[V(G) \setminus \{u, v\}]) \cup \{\{x_{\{u,v\}}, w\} \mid w \in N(\{u, v\})\}$. A graph H is called a *minor* of G , if H can be obtained from G by a sequence of edge deletions, edge contractions and vertex deletions. In a graph G , a sequence of vertices $[u_1 u_2 \dots u_\ell]$ is called a path in G , if for any $i, j \in [\ell], i \neq j, u_i \neq u_j$ and $\{u_r, u_{r+1}\} \in E(G)$ for all $r \in [\ell - 1]$. We also call the path $P = [u_1 u_2 \dots u_\ell]$ as u_1 - u_ℓ path. The internal vertices of a path $P = [u_1 u_2 \dots u_\ell]$ are $\{u_2, u_3, \dots, u_{\ell-1}\}$. A sequence of vertices $[u_1 u_2 \dots u_\ell]$ is called a cycle in G , if $u_1 = u_\ell, [u_1 u_2 \dots u_{\ell-1}]$ is a path and $\{u_{\ell-1}, u_\ell\} \in E(G)$. For a path or a cycle Q , we use $V(Q)$ to denote the set of vertices in Q . Given $k \in \mathbb{N}$, we let K_k denote the complete graph on k vertices. For a set X , we use $K[X]$ to denote the complete graph on X .

Given $a, b \in \mathbb{N}$, an $a \times b$ grid is a graph on $a \cdot b$ vertices, $v_{i,j}$ for $(i, j) \in [a] \times [b]$, such that for all $i \in [a - 1]$ and $j \in [b]$, it holds that $v_{i,j}$ and $v_{i+1,j}$ are neighbors, and for all $i \in [a]$ and $j \in [b - 1]$, it holds that $v_{i,j}$ and $v_{i,j+1}$ are neighbors. By slightly abusing the notation, for any two positive real numbers a and b , we use the term $a \times b$ grid to denote a $\lfloor a \rfloor \times \lfloor b \rfloor$ grid. For ease of presentation, for any function $f : D \rightarrow [a] \times [b]$ (where $a, b \in \mathbb{N}, i \in [a]$, and $j \in [b]$), we use $f^{-1}(i, j)$, $f^{-1}(*, j)$, and $f^{-1}(i, *)$ to denote the sets $f^{-1}((i, j))$, $f^{-1}((* , j))$, and $f^{-1}((i, *))$, respectively.

A path decomposition is defined as follows.

Definition 2.1 A *path decomposition* of a graph G is a sequence $\mathcal{P} = (X_1, X_2, \dots, X_\ell)$, where each $X_i \subseteq V(G)$ is called a *bag*, that satisfies the following conditions:

- $\bigcup_{i \in [\ell]} X_i = V(G)$.
- For every edge $\{u, v\} \in E(G)$ there exists $i \in [\ell]$ such that $\{u, v\} \subseteq X_i$.
- For every vertex $v \in V(G)$, if $v \in X_i \cap X_j$ for some $i \leq j$, then $v \in X_r$ for all $r \in \{i, \dots, j\}$.

The *width* of \mathcal{P} is $\max_{i \in [\ell]} |X_i| - 1$.

The *pathwidth* of G is the minimum width of a path decomposition of G , and it is denoted by $pw(G)$. A tree decomposition is a structure more general than a path decomposition, which is defined as follows.

Definition 2.2 A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \beta)$, where T is a tree and β is a function from $V(T)$ to $2^{V(G)}$, that satisfies the following conditions:

- $\bigcup_{x \in V(T)} \beta(x) = V(G)$.
- For every edge $\{u, v\} \in E(G)$ there exists $x \in V(T)$ such that $\{u, v\} \subseteq \beta(x)$.
- For every vertex $v \in V(G)$, if $v \in \beta(x) \cap \beta(y)$ for some $x, y \in V(T)$, then $v \in \beta(z)$ for all z on the unique path between x and y in T .

The *width* of \mathcal{T} is $\max_{x \in V(T)} |\beta(x)| - 1$. Each $\beta(x)$ is called a *bag*. Moreover, we let $\gamma(x)$ denote the union of the bags of x and its descendants.

In other words, a path decomposition is a tree decomposition where T is a path, but it will be convenient for us to think of a path decomposition as a sequence using the syntax in Definition 2.1. The *treewidth* of G is the minimum width of a tree decomposition of G , and it is denoted by $tw(G)$.

Proposition 2.3 [8] *Given a graph G and an integer k , in time $2^{O(k)} \cdot n$, we can either decide that $tw(G) > k$ or output a tree decomposition of G of width $5k$.*

A *nice tree decomposition* is a tree decomposition of a form that simplifies the design of dynamic programming (DP) algorithms. Formally,

Definition 2.4 A tree decomposition $\mathcal{T} = (T, \beta)$ of a graph G is *nice* if for the root r of T , it holds that $\beta(r) = \emptyset$, and each node $v \in V(T)$ is of one of the following types:

- **Leaf:** v is a leaf in T and $\beta(v) = \emptyset$.
- **Forget:** v has exactly one child u , and there exists a vertex $w \in \beta(u)$ such that $\beta(v) = \beta(u) \setminus \{w\}$.
- **Introduce:** v has exactly one child u , and there exists a vertex $w \in \beta(v)$ such that $\beta(v) \setminus \{w\} = \beta(u)$.
- **Join:** v has exactly two children, u and w , and $\beta(v) = \beta(u) = \beta(w)$.

Proposition 2.5 [5] *Given a graph G and a tree decomposition \mathcal{T} of G , a nice tree decomposition \mathcal{T}' of the same width as \mathcal{T} can be computed in linear time.*

Geometric Graphs. Given a set of geometric objects O , the *intersection graph* of O is a graph G with vertex set $V(G) = O$ and edge set $E(G) = \{\{u, v\} : u, v \in O, u \cap v \neq \emptyset\}$.

Let $P = \{p_1 = (x_1, y_1), p_2 = (x_2, y_2), \dots, p_n = (x_n, y_n)\}$ be a set of points in the Euclidean plane. In the *unit disk graph model*, for every $i \in [n]$, we let d_i denote the disk of radius 1 whose centre is p_i . Accordingly, we denote $D = \{d_1, d_2, \dots, d_n\}$. Then, the *unit disk graph* of D is the intersection graph of D . Alternatively, the unit disk graph of D is the geometric graph of G such that $E(G) = \{\{p_i = (x_i, y_i), p_j = (x_j, y_j)\} \mid p_i, p_j \in D, i \neq j, \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq 2\}$. In the *unit square graph model*, for every $i \in [n]$, we let s_i denote the axis-parallel unit square whose centre is p_i . Accordingly, we denote $S = \{s_1, s_2, \dots, s_n\}$. Then, the *unit square graph* of S is the intersection graph of S . Alternatively, the unit square graph of S is the geometric graph G such that $E(G) = \{\{p_i = (x_i, y_i), p_j = (x_j, y_j)\} \mid p_i, p_j \in S, i \neq j, |x_i - x_j| \leq 1, |y_i - y_j| \leq 1\}$.

3 Clique-Grid Graphs

In this section, we introduce a family of “grid-like” graphs, called clique-grid graphs, that is tailored to fit our techniques. Given a unit disk/square graph G , we extract the properties of G that we would like to exploit, and show that they can be captured by an appropriate clique-grid graph. Let us begin by giving the definition of a clique-grid graph. Roughly speaking, a graph G is a clique-grid graph if each of its vertices can be

embedded into a single cell of a grid (where multiple vertices can be embedded into the same cell), ensuring that the subgraph induced by each cell is a clique, and that each cell can interact (via edges incident to its vertices) only with cells at “distance” at most 2. Formally,

Definition 3.1 (*Clique-grid graphs*) A graph G is a *clique-grid graph* if there exists a function $f: V(G) \rightarrow [t] \times [t']$, for some $t, t' \in \mathbb{N}$, such that the following conditions are satisfied:

1. For all $(i, j) \in [t] \times [t']$, it holds that $f^{-1}(i, j)$ is a clique.
2. For all $\{u, v\} \in E(G)$, it holds that if $f(u) = (i, j)$ and $f(v) = (i', j')$ then

$$|i - i'| \leq 2 \quad \text{and} \quad |j - j'| \leq 2.$$

Such a function f is a *representation* of G .

We note that a notion similar to clique-grid graph was also used by Ito and Kadoshita [32]. For the sake of clarity, we say that a pair $(i, j) \in [t] \times [t']$ is a *cell*. Moreover, whenever we discuss a clique-grid graph, we assume that we also have the representation. Next, we show that a unit disk graph is a clique-grid graph.

Lemma 3.2 *Let D be a set of points in the Euclidean plane, and let G be the unit disk graph of D . Then, a representation f of G can be computed in polynomial time.*

Proof The proof is by creating a *grid* F containing all the points in D with horizontal and vertical lines where two consecutive horizontal/vertical lines are at a distance $\sqrt{2}$. Here each square in F is of dimension $\sqrt{2} \times \sqrt{2}$. We say that a point is in a square if it is *inside* the square or it is on the boundary of the square which is the left vertical line or the bottom horizontal line of the square. Then we define a function f which maps the points in the square to the square it belongs to. Then we prove that f is indeed a representation for G .

More formally, denote $x_{\min} = \min\{x_i \mid p_i = (x_i, y_i) \in D\}$, $x_{\max} = \max\{x_i \mid p_i = (x_i, y_i) \in D\}$, $y_{\min} = \min\{y_i \mid p_i = (x_i, y_i) \in D\}$ and $y_{\max} = \max\{y_i \mid p_i = (x_i, y_i) \in D\}$. Without loss of generality assume that $x_{\min} = 0$ and $y_{\min} = 0$ (otherwise we shift the origin to the point (x_{\min}, y_{\min})). Let $t = \lceil y_{\max}/\sqrt{2} \rceil + 1$ and $t' = \lceil x_{\max}/\sqrt{2} \rceil + 1$. Now we define $f: V(G) \rightarrow [t] \times [t']$ as follows. For all $p_i = (x_i, y_i) \in V(G)$, define $a_i = \lfloor x_i/\sqrt{2} + 1 \rfloor$, $b_i = \lfloor y_i/\sqrt{2} + 1 \rfloor$ and $f(p_i) = (a_i, b_i)$.

First, let us verify that Condition 1 in Definition 3.1 is satisfied. To this end, let $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$ be two distinct vertices in $V(G)$ such that $f(p_i) = f(p_j)$. Then, $\lfloor x_i/\sqrt{2} + 1 \rfloor = \lfloor x_j/\sqrt{2} + 1 \rfloor$ and $\lfloor y_i/\sqrt{2} + 1 \rfloor = \lfloor y_j/\sqrt{2} + 1 \rfloor$. Thus, we have that $|x_i - x_j| < \sqrt{2}$ and $|y_i - y_j| < \sqrt{2}$. In particular, $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < 2$, which implies that $(p_i, p_j) \in E(G)$.

Next, let us verify that Condition 2 in Definition 3.1 is satisfied. To this end, let $\{p_i = (x_i, y_i), p_j = (x_j, y_j)\} \in E(G)$. Recall that $f(p_i)$ and $f(p_j)$ are denoted by (a_i, b_i) and (a_j, b_j) , respectively. Thus, to prove that $f(p_j) \in \{(a', b') \mid |a_i - a'| \leq 2, |b_i - b'| \leq 2\}$, it should be shown that $|a_i - a_j| \leq 2$ and $|b_i - b_j| \leq 2$. By substituting a_i, a_j, b_i and b_j , it should be shown that

$$\begin{aligned}
 - \left\| \left\lfloor \frac{x_i}{\sqrt{2}} \right\rfloor - \left\lfloor \frac{x_j}{\sqrt{2}} \right\rfloor \right\| &\leq 2, \text{ and} \\
 - \left\| \left\lfloor \frac{y_i}{\sqrt{2}} \right\rfloor - \left\lfloor \frac{y_j}{\sqrt{2}} \right\rfloor \right\| &\leq 2.
 \end{aligned}$$

We focus on the proof of the first item, as the proof of the second item is symmetric. Without loss of generality, suppose that $x_j \leq x_i$. Then, it remains to show that $\lfloor x_i/\sqrt{2} \rfloor - \lfloor x_j/\sqrt{2} \rfloor \leq 2$. Since G is the unit disk graph of D and $\{p_i, p_j\} \in E(G)$, it holds that $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq 2$. In particular, $x_i - x_j \leq 2$. Denote $X = x_j/\sqrt{2}$. Then, $\lfloor x_i/\sqrt{2} \rfloor - \lfloor x_j/\sqrt{2} \rfloor \leq \lfloor X + \sqrt{2} \rfloor - \lfloor X \rfloor \leq 2$. \square

Similarly, we can show the following.

Lemma 3.3 *Let S be a set of points in the Euclidean plane, and let G be the unit square graph of S . Then, a representation f of G can be computed in polynomial time.*

Proof (Sketch) Like the case of Lemma 3.2, the proof is by creating a grid F containing all the points in D . But here the grid is formed by horizontal and vertical lines where two consecutive horizontal/vertical lines are at a distance 1. Here each square in F is of dimension 1×1 . Again, we say that a point is in a square if it is *inside* the square or it is on the boundary of the square which is the left vertical line or the bottom horizontal line of the square. Then the required function f maps the points in the square to the square it belongs to. \square

Consequently, we have the following.

Corollary 3.4 *Let (G, O, H, k) ($((G, O, k))$) be an instance of SI (resp. LONGEST CYCLE) on unit disk/square graphs. Then, in polynomial time, one can output a representation f such that (G, f, H, k) (resp. (G, f, k)) is an instance of SI (resp. LONGEST CYCLE) on clique-grid graphs that is equivalent to (G, O, H, k) (resp. (G, O, k)).*

We conclude this section by introducing the definition of an ℓ -NCTD, which is useful for doing our dynamic programming algorithms.

Definition 3.5 A tree decomposition $\mathcal{T} = (T, \beta)$ of a clique-grid graph G with representation f is a *nice ℓ -clique tree decomposition*, or simply an ℓ -NCTD, if for the root r of T , it holds that $\beta(r) = \emptyset$, and for each node $v \in V(T)$, it holds that

- There exist at most ℓ cells, $(i_1, j_1), \dots, (i_\ell, j_\ell)$, such that $\beta(v) = \bigcup_{t=1}^{\ell} f^{-1}(i_t, j_t)$, and
- The node v is of one of the following types:
 - **Leaf:** v is a leaf in T and $\beta(v) = \emptyset$.
 - **Forget:** v has exactly one child u , and there exists a cell $(i, j) \in [t] \times [t']$ such that $f^{-1}(i, j) \subseteq \beta(u)$ and $\beta(v) = \beta(u) \setminus f^{-1}(i, j)$.
 - **Introduce:** v has exactly one child u , and there exists a cell $(i, j) \in [t] \times [t']$ such that $f^{-1}(i, j) \subseteq \beta(v)$ and $\beta(v) \setminus f^{-1}(i, j) = \beta(u)$.
 - **Join:** v has exactly two children, u and w , and $\beta(v) = \beta(u) \cup \beta(w)$.

A nice ℓ -clique path decomposition, or simply an ℓ -NCPD, is an ℓ -NCTD where T is a path. In this context, for convenience, we use the notation referring to a sequence presented in Sect. 2.

4 The Cell Graph of a Clique-Grid Graph

In this section, we introduce two compact representations of clique-grid graphs. By examining these representations, we are able to infer information on the structure of clique-grid graphs that are also unit disk/square graphs.

Definition 4.1 (*Backbone*) Given a clique-grid graph G with representation $f : V(G) \rightarrow [t] \times [t']$, an induced subgraph H of G is a *backbone* for (G, f) if for every two distinct cells $L, L' \in [t] \times [t']$ for which there exist $u \in f^{-1}(L)$ and $v \in f^{-1}(L')$ such that $\{u, v\} \in E(G)$, there also exist $u' \in f^{-1}(L)$ and $v' \in f^{-1}(L')$ such that $\{u', v'\} \in E(H)$. If no induced subgraph of H is a backbone for (G, f) , then H is a *minimal backbone* for (G, f) .

First, we bound the maximum degree of a minimal backbone.

Lemma 4.2 *Let G be a clique-grid graph with representation f , and let H be a minimal backbone for (G, f) . Then, for all $(i, j) \in [t] \times [t']$, it holds that $|f^{-1}(i, j) \cap V(H)| \leq 24$. Furthermore, the maximum degree of H is at most 599.*

Proof By Condition 2 in Definition 3.1, we have that for all cells $(i, j) \in [t] \times [t']$, it holds that $|f^{-1}(i, j) \cap V(H)| \leq |\{(i', j') \in [t] \times [t'] \setminus \{(i, j)\} : |i - i'| \leq 2 \text{ and } |j - j'| \leq 2\}| \leq 24$. Thus, for all $(i, j) \in [t] \times [t']$, the degree in H of a vertex in $f^{-1}(i, j) \cap V(H)$ is bounded by

$$\begin{aligned} & \left| \left(\bigcup_{\substack{(i', j') \in [t] \times [t'] \\ |i - i'| \leq 2 \\ |j - j'| \leq 2}} f^{-1}(i, j) \cap V(H) \right) \setminus \{v\} \right| \\ & \leq |\{(i', j') \in [t] \times [t'] : |i - i'| \leq 2 \text{ and } |j - j'| \leq 2\}| \cdot 24 - 1 \\ & = 25 \cdot 24 - 1 = 599. \end{aligned} \quad \square$$

Note that it is easy to compute a minimal backbone. The most naive computation simply initializes $H = G$; then, for every vertex $v \in V(G)$, it checks if the graph $H \setminus \{v\}$ has the same backbone as H , in which case it updates H to $H \setminus \{v\}$. Thus, we have the following.

Observation 4.3 *Given a clique-grid graph G with representation f , a minimal backbone H for (G, f) can be computed in polynomial time.*

To analyze the treewidth of a backbone, we need the following.¹

¹ The paper [23] does not consider unit square graphs, but the arguments it presents for unit disk graphs can be adapted to handle unit square graphs as well.

Proposition 4.4 [23] *Any unit disk/square graph with maximum degree Δ contains a $\frac{tw}{100\Delta^3} \times \frac{tw}{100\Delta^3}$ grid as a minor, where tw is the treewidth of the graph.*

Thus, we have the following.

Lemma 4.5 *Given a clique-grid graph G that is a unit disk/square graph, a representation f of G and an integer $\ell \in \mathbb{N}$, in time $2^{O(\ell)} \cdot n^{O(1)}$, one can either correctly conclude that G contains an $\alpha\ell \times \alpha\ell$ grid as a minor, or obtain a minimal backbone H for (G, f) with a nice tree decomposition \mathcal{T} of width at most 5ℓ , where $\alpha = \frac{1}{100 \cdot 599^3}$.*

Proof By Lemma 4.2, Observation 4.3 and Proposition 4.4, in polynomial time, one can compute a minimal backbone of H such that H either contains a $\alpha\ell \times \alpha\ell$ grid as a minor or has treewidth at most ℓ . Since H is a subgraph of G , it holds that if H contains an $a \times b$ grid as a minor, then G also contains an $a \times b$ grid as a minor. Thus, by Propositions 2.5 and 2.3, we conclude the proof of the lemma. \square

Next, we define a more compact representation of a clique-grid graph.

Definition 4.6 (*Cell graph*) *Given a clique-grid graph G with representation $f: V(G) \rightarrow [t] \times [t']$, the cell graph of G , denoted by $cell(G)$, is the graph on the vertex-set $\{v_{i,j} : i \in [t], j \in [t'], f^{-1}(i, j) \neq \emptyset\}$ and edge-set $\{\{v_{i,j}, v_{i',j'}\} : (i, j) \neq (i', j'), \exists u \in f^{-1}(i, j) \exists v \in f^{-1}(i', j') \text{ such that } \{u, v\} \in E(G)\}$.*

By Definitions 4.1 and 4.6, we directly conclude the following.

Observation 4.7 *For a clique-grid graph G , a representation f of G and a backbone H for (G, f) , it holds that $cell(G)$ is a minor of H .*

Since for any graph G and a minor H of G , it holds that $tw(H) \leq tw(G)$, we have the following.

Observation 4.8 *For a clique-grid graph G , a representation f of G and a backbone H for (G, f) , it holds that $tw(cell(G)) \leq tw(H)$.*

Overall, from Lemma 4.5 and Observation 4.8, we directly have the following.

Lemma 4.9 *Given a clique-grid graph G that is a unit disk/square graph, a representation f of G and an integer $\ell \in \mathbb{N}$, in time $2^{O(\ell)} \cdot n^{O(1)}$, one can either correctly conclude that G contains an $\alpha\ell \times \alpha\ell$ grid as a minor, or compute a nice tree decomposition of $cell(G)$ of width at most 5ℓ , where $\alpha = \frac{1}{100 \cdot 599^3}$.*

Note that a nice tree decomposition of $cell(G)$ of width 5ℓ corresponds to a 5ℓ -NCTD of G . In other words, Lemma 4.9 implies the following.

Corollary 4.10 *Given a clique-grid graph G that is a unit disk/square graph, a representation f of G and an integer $\ell \in \mathbb{N}$, in time $2^{O(\ell)} \cdot n^{O(1)}$, one can either correctly conclude that G contains an $\alpha\ell \times \alpha\ell$ grid as a minor, or compute a 5ℓ -NCTD of G , where $\alpha = \frac{1}{100 \cdot 599^3}$.*

5 Turing Kernels

For the sake of uniformity, throughout this section, we denote an instance (G, O, k) ((G, f, k)) of LONGEST CYCLE on unit disk/square graphs (resp. clique-grid graphs) also by (G, O, H, k) (resp. (G, f, H, k)) where H is the empty graph. Our objective is to show that both SI and LONGEST CYCLE on unit disk/square graphs admit a Turing kernel [13, Chap. 9]. More precisely, we prove the following.

Theorem 5.1 *Let (G, O, H, k) be an instance of SI (LONGEST CYCLE) on unit disk/square graphs. Then, in polynomial time, one can output a set \mathcal{I} of instances of SI (resp. LONGEST CYCLE) on unit disk/square graphs such that (G, O, H, k) is a YES-instance if and only if at least one instance in \mathcal{I} is a YES-instance, and for all $(\widehat{G}, \widehat{O}, \widehat{H}, \widehat{k}) \in \mathcal{I}$, it holds that $|V(\widehat{G})| = \mathcal{O}(k^3)$, $|E(\widehat{G})| = \mathcal{O}(k^4)$, $\widehat{H} = H$ and $\widehat{k} = k$.*

To prove Theorem 5.1, we first need two definitions.

Definition 5.2 Let G be a clique-grid graph with representation $f : V(G) \rightarrow [t] \times [t']$, H' be a subgraph of G , and $\ell \in \mathbb{N}$. We say that H' is ℓ -stretched if there exist cells $(i, j), (i', j') \in [t] \times [t']$ such that the following conditions are satisfied:

- It holds that $|i - i'| \geq 2\ell$ or $|j - j'| \geq 2\ell$ (or both).
- It holds that $V(H') \cap f^{-1}(i, j) \neq \emptyset$ and $V(H') \cap f^{-1}(i', j') \neq \emptyset$.

For a clique-grid graph G with representation $f : V(G) \rightarrow [t] \times [t']$ and a subgraph G' of G we denote $i_{\min}(G') = \min\{i \in [t] : (\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(G') \neq \emptyset\}$, $i_{\max}(G') = \max\{i \in [t] : (\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(G') \neq \emptyset\}$, $j_{\min}(G') = \min\{j \in [t'] : (\bigcup_{i \in [t]} f^{-1}(i, j)) \cap V(G') \neq \emptyset\}$ and $j_{\max}(G') = \max\{j \in [t'] : (\bigcup_{i \in [t]} f^{-1}(i, j)) \cap V(G') \neq \emptyset\}$. We proceed by proving two claims concerning solutions to LONGEST CYCLE and SI on clique-grid graphs.

Lemma 5.3 *Let $I = (G, f : V(G) \rightarrow [t] \times [t'], H, k)$ be an instance of SI on clique-grid graphs. Then, for any subgraph H' of G that is isomorphic to H , it holds that H' is not $2k$ -stretched.*

Proof Let H' be a subgraph of G that is isomorphic to H . To prove that H' is not $2k$ -stretched, we need to prove that $i_{\max}(H') - i_{\min}(H') < 2k$ and $j_{\max}(H') - j_{\min}(H') < 2k$. We only prove that $i_{\max}(H') - i_{\min}(H') < 2k$, as the proof that $j_{\max}(H') - j_{\min}(H') < 2k$ is symmetric.

Let $i_1 < i_2 < \dots < i_\ell$ for the appropriate ℓ be the set of indices $i \in [t]$ such that $(\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(H') \neq \emptyset$. Note that $i_1 = i_{\min}(H')$ and $i_\ell = i_{\max}(H')$. We claim that for all $r \in [\ell - 1]$, it holds that $i_{r+1} - i_r \leq 2$. Suppose, by way of contradiction, that there exists $r \in [\ell - 1]$ such that $i_{r+1} - i_r > 2$. Recall that H is a connected graph, and therefore H' is also a connected graph. Thus, there exists an edge $\{u, v\} \in E(H') \subseteq E(G)$ and indices $i \leq i_r$ and $i' \geq i_{r+1}$ such that $u \in \bigcup_{j \in [t']} f^{-1}(i, j)$ and $v \in \bigcup_{j \in [t']} f^{-1}(i', j)$. However, this contradicts the fact that f is a representation of G . Also, since $|V(H')| \leq k$ and H' is connected, $i_{\max}(H') - i_{\min}(H') < 2k$. □

Definition 5.4 Let $I = (G, f: V(G) \rightarrow [t] \times [t'], k)$ be an instance of LONGEST CYCLE on clique-grid graphs. We say that I is a *stretched instance* if G has a cycle C that is ℓ -stretched for some $\ell \geq 2k$.

Lemma 5.5 Let $I = (G, f: V(G) \rightarrow [t] \times [t'], k)$ be an instance of LONGEST CYCLE on clique-grid graphs. Then, it can be determined in polynomial time whether I is a stretched instance, in which case it is also a YES-instance.

Proof It is well known that for any given graph and pair of vertices in this graph, one can determine (in polynomial time) whether the given graph has a cycle that contains both given vertices by checking whether they are in the same biconnected component of the graph (see, e.g., [29]). Thus, by considering every pair (u, v) of vertices in $V(G)$ such that $|i - i'| \geq 2k$ or $|j - j'| \geq 2k$ (or both) where $f(u) = (i, j)$ and $f(v) = (i', j')$, we can determine (in polynomial time) whether I is a stretched instance.

Now, suppose that I is a stretched instance. Then, G has a cycle C that is ℓ -stretched for some $\ell \geq 2k$. Note that $I' = (G, f, C, |V(C)|)$ is a YES-instance of SI on clique-grid graphs. Thus, by Lemma 5.3, it holds that C is not $2|V(C)|$ -stretched. Therefore, $\ell < 2|V(C)|$, and since $\ell \geq 2k$, we conclude that $k < |V(C)|$. Thus, I is a YES-instance. \square

Next, we prove Theorem 5.1. Our method is inspired by Baker's technique [3].

Proof of Theorem 5.1 First, by Corollary 3.4, we obtain (in polynomial time) an instance (G, f, H, k) of SI (LONGEST CYCLE) on clique-grid graphs that is equivalent to (G, O, H, k) . Now in polynomial time, we output a set \mathcal{I}' of instances of SI (resp. LONGEST CYCLE) on clique-grid graphs such that (G, f, H, k) (and hence (G, O, H, k)) is a YES-instance if and only if at least one instance in \mathcal{I}' is a YES-instance, and for all $(\widehat{G}, \widehat{f}, \widehat{H}, \widehat{k}) \in \mathcal{I}'$, it holds that \widehat{G} is either an induced subgraph of G or $K_{\widehat{k}}$, $\widehat{t}, \widehat{t}' \leq 2k = \mathcal{O}(\widehat{k})$, $|\widehat{f}^{-1}(i, j)| < \widehat{k}$ for any cell $(i, j) \in [\widehat{t}] \times [\widehat{t}']$, $\widehat{H} = H$ and $\widehat{k} = k$.

Towards constructing such a set \mathcal{I}' , suppose that there exists a cell $(i, j) \in [t] \times [t']$ such that $|f^{-1}(i, j)| \geq k$, then by Definition 3.1, $G[f^{-1}(i, j)]$ is a clique on at least k vertices. In particular, the pattern H is a subgraph of $G[f^{-1}(i, j)]$, and therefore it is also a subgraph of G . Thus, in this case, we conclude the proof by setting \mathcal{I}' to be the set that contains only one instance, $(K_k, \widehat{f}: V(K_k) \rightarrow [1] \times [1], H, k)$. From now on, suppose that for all cells $(i, j) \in [t] \times [t']$, it holds that $|f^{-1}(i, j)| < k$.

Second, in case the input instance I is of LONGEST CYCLE, we use the computation given by Lemma 5.5 to determine whether I is a stretched instance. If the answer is positive, then by Lemma 5.5, it holds that I is a YES-instance. In this case, we again conclude the proof by setting \mathcal{I}' to be the set that contains only one instance, $(K_k, \widehat{f}: V(K_k) \rightarrow [1] \times [1], H, k)$. From now on, also suppose that if the input instance I is of LONGEST CYCLE, then it is not stretched.

Now, our algorithm to construct \mathcal{I}' works as follows. For every $(p, q) \in [t] \times [t']$, it computes

$$G_{p,q} = G \left[\bigcup_{\substack{p \leq i < \min\{p+2k, t+1\} \\ q \leq j < \min\{q+2k, t'+1\}}} f^{-1}(i, j) \right].$$

Let $f_{p,q} : V(G_{p,q}) \rightarrow [2k] \times [2k]$ be such that $f_{p,q}(v) := (i - p + 1, j - q + 1)$, where $(i, j) = f(v)$. Thus, since f is a representation of G , it holds that $f_{p,q}$ is a representation of $G_{p,q}$. Finally, our algorithm outputs $\mathcal{I}' = \{I_{p,q} = (G_{p,q}, f_{p,q}, H, k) : (p, q) \in [t] \times [t']\}$.

Next, we show that (G, f, H, k) is a YES-instance if and only if at least one instance in \mathcal{I}' is a YES-instance. Since for all $(G_{p,q}, f_{p,q}, H, k) \in \mathcal{I}'$, it holds that $G_{p,q}$ is an induced subgraph of G , we have that if (G, f, H, k) is a NO-instance, then every instance in \mathcal{I}' is NO-instance as well. Next, suppose that (G, f, H, k) is a YES-instance. Let us consider two cases.

- (G, f, H, k) is an instance of SI. Then, let H' be a subgraph of G that is isomorphic to H . By Lemma 5.3, it holds that both $i_{\max}(H') - i_{\min}(H') < 2k$ and $j_{\max}(H') - j_{\min}(H') < 2k$. Therefore, H' is a subgraph of $G_{p,q}$, where $p = i_{\min}(H')$ and $q = j_{\min}(H')$. This implies that $I_{p,q}$ is a YES-instance.
- (G, f, H, k) is an instance of LONGEST CYCLE. Then, let C be a subgraph of G that is a cycle on at least k vertices. Since (G, f, H, k) is not stretched, it holds that both $i_{\max}(C) - i_{\min}(C) < 2k$ and $j_{\max}(C) - j_{\min}(C) < 2k$. Therefore, C is a subgraph $G_{p,q}$, where $p = i_{\min}(C)$ and $q = j_{\min}(C)$. This implies that $I_{p,q}$ is a YES-instance.

To conclude our proof, it is sufficient to show that for any instance $(\widehat{G}, \widehat{f}, \widehat{H}, \widehat{k}) \in \mathcal{I}'$, we can compute (in polynomial time) an equivalent instance (G', O', H', k') of SI (LONGEST CYCLE) on unit disk/square graphs such that $|O'| = \mathcal{O}(k'^3)$, $H' = H$ and $k' = k$. To this end, fix some instance $(\widehat{G}, \widehat{f}, \widehat{H}, \widehat{k}) \in \mathcal{I}'$. Let us first handle the simple case where \widehat{G} is equal to $K_{\widehat{k}}$. Here, we conclude the proof by defining $p'_i = (0, i/k)$ for $i \in [k]$, and then setting $O' = \{p'_i : i \in [k]\}$.

Now, suppose that \widehat{G} is an induced subgraph of G . Then, we have that $V(\widehat{G}) \subseteq O$, and the unit disk/square graph of $V(\widehat{G})$ is exactly \widehat{G} . Thus, we define $(G', O', H', k') = (\widehat{G}, V(\widehat{G}), \widehat{H}, \widehat{k})$. It remains to show that $|V(\widehat{G})| = \mathcal{O}(k^3)$ and $|E(\widehat{G})| = \mathcal{O}(k^4)$.

By Definition 3.1, it holds that

$$|V(\widehat{G})| = \sum_{(i,j) \in [\widehat{t}] \times [\widehat{t}']} |f^{-1}(i, j)|.$$

Thus, since $\widehat{t}, \widehat{t}' = \mathcal{O}(k)$ and $|f^{-1}(i, j)| \leq k$ for $(i, j) \in [\widehat{t}] \times [\widehat{t}']$, we have that $|V(\widehat{G})| = \mathcal{O}(k^3)$.

Now, denote $X = \{((i, j), (i', j')) : i, i' \in [t], j, j' \in [t'], |i - i'| \leq 2, |j - j'| \leq 2\}$. Since $\widehat{t}, \widehat{t}' = \mathcal{O}(k)$, we have that $|X| = \mathcal{O}(k^2)$. By Definition 3.1, it also holds that

$$|E(\widehat{G})| \leq \sum_{((i,j),(i',j')) \in X} |f^{-1}(i, j)| \cdot |f^{-1}(i', j')|.$$

Thus, since $|X| = \mathcal{O}(k^2)$, and $|\widehat{f}^{-1}(i, j)| \leq k$ for $(i, j) \in \widehat{[t]} \times \widehat{[t']}$, we have that $|E(\widehat{G})| = \mathcal{O}(k^4)$. □

We obtain the following corollary.

Corollary 5.6 *Let (G, O, H, k) be an instance of SI (LONGEST CYCLE) on unit disk/square graphs. Then, in polynomial time, one can output a set \mathcal{I} of instances of SI (resp. LONGEST CYCLE) on clique-grid graphs such that (i) (G, O, H, k) is a YES-instance if and only if at least one instance in \mathcal{I} is a YES-instance, and (ii) for all $(\widehat{G}, \widehat{f}: V(\widehat{G}) \rightarrow \widehat{[t]} \times \widehat{[t']}, \widehat{H}, \widehat{k}) \in \mathcal{I}$, it holds that \widehat{G} is either an induced subgraph of G or $K_{\widehat{k}}$, $|\widehat{f}^{-1}(L)| < \widehat{k}$ for any cell $L \in \widehat{[t]} \times \widehat{[t']}$, $\widehat{H} = H$, $\widehat{k} = k$ and $\widehat{t}, \widehat{t'} \leq 2k$.*

The proof of Corollary 5.6 follows from the proof of Theorem 5.1, even though a direct application of Corollary 3.4 on each output instance of Theorem 5.1 implies Corollary 5.6 without explicit bound on t and t' . Corollary 5.6 is used in Sects. 6 and 7.

6 Exact k -Cycle

In this section we prove the following theorem.

Theorem 6.1 *EXACT k -CYCLE on unit disk/square graphs can be solved in $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ time.*

Because of Corollary 5.6, towards proving Theorem 6.1, we can assume that the input to our algorithm is $(G, f: V(G) \rightarrow [t] \times [t'], k)$ where G is a clique-grid graph with a representation f , $|f^{-1}(L)| < k$ for all $L \in [t] \times [t']$ and $t, t' \leq 2k$. Without loss of generality we can assume that f is a function from $V(G)$ to $[2k] \times [2k]$, because $[t] \times [t'] \subseteq [2k] \times [2k]$. Since $f^{-1}(L) < k$ for all $L \in [t] \times [t']$ the number of vertices in the input graph is bounded by $\mathcal{O}(k^2)$.

Given an instance $(G, f: V(G) \rightarrow [2k] \times [2k], k)$, the algorithm applies a method inspired by Baker’s technique [3] and obtains a family \mathcal{F} of $2^{\mathcal{O}(\sqrt{k} \log k)}$ instances of EXACT k -CYCLE. The family \mathcal{F} has the following properties:

1. In each of these instances the input graph is an induced subgraph of G and has size $k^{\mathcal{O}(1)}$.
2. The input $(G, f: V(G) \rightarrow [2k] \times [2k], k)$ is a YES-instance if and only if there exists an instance $(H, f^*: V(H) \rightarrow [2k] \times [2k], k) \in \mathcal{F}$ which is a YES-instance.
3. Moreover, for any instance $(H, f^*: V(H) \rightarrow [2k] \times [2k], k) \in \mathcal{F}$, H has a nice $7\sqrt{k}$ -clique path decomposition ($7\sqrt{k}$ -NCPD).

We will call the family \mathcal{F} satisfying the above properties as *good family*. Let $(H, f^*: V(H) \rightarrow [2k] \times [2k], k)$ be an instance of \mathcal{F} . Let $\mathcal{P} = (X_1, \dots, X_q)$ be a $7\sqrt{k}$ -NCPD of H . We first prove that if there is a cycle of length k in H , then there is a cycle C of length k in H such that for any two distinct cells L and L' of f , the number of edges with one endpoint in L and other in L' is at most 5. Let C be such a cycle in H . Then using the property of C we get the following important property:

For any $i \in [q]$, the number of edges of C with one endpoint in X_i and other in $\bigcup_{i < j \leq q} X_j$ is upper bounded by $\mathcal{O}(\sqrt{k})$.

The above mentioned property allows us to design a dynamic programming (DP) algorithm over the $7\sqrt{k}$ -NCPD, \mathcal{P} , for EXACT k -CYCLE in time $2^{\mathcal{O}(\sqrt{k} \log k)}$. Now we are ready to give formal details about the algorithm.

Lemma 6.2 *Let $(G, f: V(G) \rightarrow [2k] \times [2k], k)$ be an instance of EXACT k -CYCLE, where G is a clique-grid graph with representation f , $|f^{-1}(L)| < k$ for all $L \in [2k] \times [2k]$. Given $(G, f: V(G) \rightarrow [2k] \times [2k], k)$, there is an algorithm running in time $2^{\mathcal{O}(\sqrt{k} \log k)}$ that outputs a good family \mathcal{F} .*

Proof Let C be a cycle of length k in G . First we define a column of the $2k \times 2k$ grid. For any $j \in [2k]$ the set of cells $\{(i, j) : i \in [2k]\}$ is called a column. There are $2k$ columns for the $2k \times 2k$ grid. We partition them into k blocks of two consecutive columns and label them from the set of labels $[\lceil \sqrt{k} \rceil]$. That is, each pair of columns $2i - 1$ and $2i$, where $i \in [k]$ is labelled with $((i - 1) \bmod \lceil \sqrt{k} \rceil) + 1$. Then by the pigeonhole principle there is a label $\ell \in \{1, 2, \dots, \lceil \sqrt{k} \rceil\}$ such that the number of vertices from $V(C)$ which are in columns labelled ℓ is at most \sqrt{k} . As $|V(G)| \leq k^{\mathcal{O}(1)}$, the number of vertices of G in columns labelled ℓ is at most $k^{\mathcal{O}(1)}$. We guess the vertices of $V(C)$ which are in the columns labelled ℓ . The number of potential guesses is bounded by $k^{\mathcal{O}(\sqrt{k})}$. Let Y be the set of guessed vertices of $V(C)$ which are in the columns labelled by ℓ . Notice that $|Y| \leq \sqrt{k}$. Then we delete all the vertices in columns labelled ℓ , except the vertices of Y . Let S be the set of deleted vertices. By the property 2 of clique-grid graph, $G \setminus (S \cup Y)$ is a disjoint union of clique-grid graphs each of which is represented by a function with at most $2(\lceil \sqrt{k} \rceil - 1) \leq 2\sqrt{k}$ columns. That is, let $G_1 = G[\bigcup_{j=1}^{2(\ell-1)} f^{-1}(*, j)]$, and $G_{i+1} = G[\bigcup_{j=(i-1)2\lceil \sqrt{k} \rceil+2\ell+1}^{\min\{(i-2)\lceil \sqrt{k} \rceil+2\ell-2, 2k\}} f^{-1}(*, j)]$ for all $i \in \{1, \dots, \lceil \sqrt{k} \rceil\}$.

By property 2 of clique-grid graph, $G \setminus (S \cup Y) = G_1 \uplus \dots \uplus G_{\lceil \sqrt{k} \rceil+1}$.

Claim 1 $G \setminus S$ has a nice $7\sqrt{k}$ -clique path decomposition ($7\sqrt{k}$ -NCPD).

Proof First, for each $i \in \{1, \dots, \lceil \sqrt{k} \rceil + 1\}$, we define a path decomposition of G_i (in the next paragraph) such that each bag is a union of at most $6\sqrt{k}$ many cells of f_i . As $G \setminus (S \cup Y) = G_1 \uplus \dots \uplus G_{\lceil \sqrt{k} \rceil+1}$, and $|Y| \leq \sqrt{k}$, by adding Y to each bag of all path decompositions we can get a required nice $7\sqrt{k}$ -clique path decomposition for $G \setminus S$.

Now, for each G_i , we define a path decomposition $\mathcal{P}_i = (X_{i,1}, X_{i,2}, \dots, X_{i,2k-2})$ where $X_{i,j} = f_i^{-1}(j, *) \cup f_i^{-1}(j + 1, *) \cup f_i^{-1}(j + 2, *)$. We claim that \mathcal{P}_i is indeed a path decomposition of G_i . Notice that $\bigcup_{j=1}^{k-1} X_{i,j} = f_i^{-1}(*, *) = V(G_i)$. By property 2 of clique-grid graph, we have that for each edge $\{u, v\} \in E(G)$, there exists $j \in [2k - 2]$ such that $\{u, v\} \in X_{i,j}$. For each $u \in V(G)$, u is contained in at most three bags and these bags are consecutive in the sequence $(X_{i,1}, X_{i,2}, \dots, X_{i,2k-2})$. Hence \mathcal{P}_i is a path decomposition of G_i . Since $X_{i,j} = f_i^{-1}(j, *) \cup f_i^{-1}(j + 1, *) \cup f_i^{-1}(j + 2, *)$, the number of columns in f_i is at

most $2\sqrt{k}$ and each cell of f_i is a cell of f , we have that each $X_{i,j}$ is a union of $6\sqrt{k}$ many cells of f . Since $G \setminus (S \cup Y) = G_1 \uplus \dots \uplus G_{\lceil\sqrt{k}\rceil+1}$, the sequence $\mathcal{P}' = (X_{1,1}, \dots, X_{1,2k-2}, X_{2,1}, \dots, X_{2,2k-2}, \dots, X_{\lceil\sqrt{k}\rceil+1,1}, \dots, X_{\lceil\sqrt{k}\rceil+1,2k-2})$ is a path decomposition of $G \setminus (S \cup Y)$. Moreover, each bag is a union of vertices from at most $6\sqrt{k}$ cells of f . Also, since $|Y| \leq \sqrt{k}$, the sequence $\mathcal{P} = (X_{1,1} \cup Y, \dots, X_{1,k-2} \cup Y, \dots, X_{\lceil\sqrt{k}\rceil+1,1}, \dots, X_{\lceil\sqrt{k}\rceil+1,k-2} \cup Y)$ obtained by adding Y to each bag of \mathcal{P}' we get a path decomposition of $G \setminus S$. Moreover, each bag in \mathcal{P} is a union of vertices from at most $7\sqrt{k}$ cells of f . We can turn the path decomposition \mathcal{P} to a $7\sqrt{k}$ -NCPD by an algorithm similar to the one mentioned in Proposition 2.5. \square

Our algorithm will construct a family \mathcal{F} as follows. For each $\ell \in \{1, \dots, \lceil\sqrt{k}\rceil\}$ and for two subsets of vertices S and Y such that $S \cup Y$ is a set of vertices in the columns labelled ℓ and $|Y| \leq \sqrt{k}$, our algorithm will include an instance $(G \setminus S, f|_{V(G)\setminus S}, k)$ in \mathcal{F} . The number of choices of S and Y is bounded by $2^{\mathcal{O}(\sqrt{k} \log k)}$ and thus the size of \mathcal{F} is bounded by $2^{\mathcal{O}(\sqrt{k} \log k)}$.

We claim that \mathcal{F} is indeed a good family. Suppose there is a cycle C of length k in G . Then, by the pigeonhole principle there is $\ell \in \{1, \dots, \lceil\sqrt{k}\rceil\}$ such that at most \sqrt{k} vertices from $V(C)$ are in the columns labelled by ℓ . Let S' be the set of vertices in the columns labelled by ℓ . Let $Y = S' \cap V(C)$ and $S = S' \setminus Y$. Notice that $|Y| \leq \sqrt{k}$. The instance $(G \setminus S, f|_{V(G)\setminus S}, k)$ in \mathcal{F} , is a YES-instance. This completes the proof of the lemma. \square

Now we can assume that we are solving EXACT k -CYCLE on (H, f, k) , where $(H, f, k) \in \mathcal{F}$ (here we rename the function $f|_{V(H)}$ with f for ease of presentation). In our algorithm we seek for a special kind of cycle which is defined as follows.

Definition 6.3 Let H be a clique-grid graph with representation f . A cycle C in H is called a simplified cycle if for any two cells L and L' of f , the number of edges of $E(C)$ with one endpoint in L and other in L' is at most 5.

Now we prove that if there is a cycle of length k in H , then there is a simplified cycle of length k in H . Towards that we first define some notations. For a path $P = [u_1 u_2 \dots u_\ell]$, we use \overleftarrow{P} to denote the path $[u_\ell u_{\ell-1} \dots u_1]$. For any two paths $P_1 = [u_1 \dots u_i]$ and $P_2 = [u_i \dots u_\ell]$, we use $P_1 P_2$ to denote the path $[u_1 u_2 \dots u_\ell]$.

Lemma 6.4 Let $(H, f: V(H) \rightarrow [2k] \times [2k], k)$ be a YES-instance of EXACT k -CYCLE. Then there is a simplified cycle of length k in H .

Proof Let C be a cycle of length k such that the number edges of $E(C)$ whose endpoints are in different cells is minimized. We claim that C is a simplified cycle in H . Suppose not. Then there exist two cells L and L' such that the number of edges of $E(C)$ with one endpoint in L and other in L' is at least 6. Let $C = P_1[u_1 v_1]P_2[u_2 v_2]P_3[u_3 v_3]P_4[u_4 v_4]P_5[u_5 v_5]P_6[u_6 v_6]$ where for each $\{u_r, v_r\}, r \in \{1, \dots, 6\}$, one endpoint is in the cell L and other is in the cell L' , and each subpath $P_\ell, \ell \in \{1, \dots, 6\}$, can be empty too. Since C is a cycle, at least 3 edges from $\{\{u_r, v_r\} : r \in \{1, \dots, 6\}\}$ form a matching. Let $\{u_{r_1}, v_{r_1}\}, \{u_{r_2}, v_{r_2}\}$ and $\{u_{r_3}, v_{r_3}\}$ be a matching of size 3, where $\{r_1, r_2, r_3\} \subseteq \{1, \dots, 6\}$. Then, by

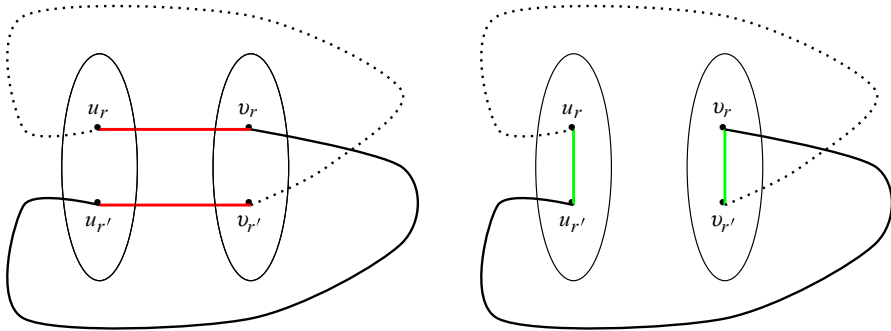


Fig. 1 Illustration of Lemma 6.4. Figure on the left is the cycle $C = [u_r v_r]Q_1[u_{r'} v_{r'}]Q_2$ and the one on the right is the cycle $C' = [u_r u_{r'}]\overleftarrow{Q}_1[v_r v_{r'}]Q_2$

the pigeonhole principle there exist $r, r' \in \{r_1, r_2, r_3\}$ such that either $u_r, u_{r'} \in f^{-1}(L)$ or $u_r, u_{r'} \in f^{-1}(L')$. Without loss of generality assume that $u_r, u_{r'} \in f^{-1}(L)$ (otherwise we rename cell L with L' and vice versa). That is, $C = [u_r v_r]Q_1[u_{r'} v_{r'}]Q_2$ such that $u_r, u_{r'} \in f^{-1}(L)$ and $v_r, v_{r'} \in f^{-1}(L')$. Then, since $f^{-1}(L)$ and $f^{-1}(L')$ are cliques, $C' = [u_r u_{r'}]\overleftarrow{Q}_1[v_r v_{r'}]Q_2$ is a cycle of length k in H , such that the number edges of $E(C')$ whose endpoints are in different cells is less than that of $E(C)$, which is a contradiction to our assumption. See Fig. 1 for an illustration of C and C' . This completes the proof of the lemma. \square

Next we design a DP algorithm that finds a simplified cycle of length k , if it exists.

Lemma 6.5 *Let $(H, f: V(H) \rightarrow [2k] \times [2k], k) \in \mathcal{F}$ be an instance of EXACT k -CYCLE and \mathcal{P} be a $7\sqrt{k}$ -NCPD of H . Then, given $(H, f: V(H) \rightarrow [2k] \times [2k], k)$ and \mathcal{P} , there is an algorithm \mathcal{A} which runs in time $2^{\mathcal{O}(\sqrt{k} \log k)}$, and outputs YES, if there is simplified cycle of length k in H . Otherwise algorithm \mathcal{A} will output NO.*

Proof Algorithm \mathcal{A} is a DP algorithm over the $7\sqrt{k}$ -NCPD $\mathcal{P} = (X_1, \dots, X_q)$ of H . For any $\ell \in [q]$, we define H_ℓ to be the induced subgraph $H[\bigcup_{i \leq \ell} X_i]$ of H . Define \mathcal{C} to be the set of all simplified cycles of length k in H . Let $C \in \mathcal{C}$. Since \mathcal{P} is a $7\sqrt{k}$ -NCPD and the fact that for any two distinct cells L and L' of f , the number of edges of C with one endpoint in L and other in L' is at most 5 (because C is a simplified cycle in H), we have that for any bag X_ℓ of \mathcal{P} , the number of vertices of $V(C) \cap X_\ell$ which has a neighbor in $V(H) \setminus X_\ell$ is bounded by $\mathcal{O}(\sqrt{k})$. This allows us to keep only $2^{\mathcal{O}(\sqrt{k} \log k)}$ states in the DP algorithm.

For a set Q of paths (of length 0 or more) and cycles, we define $\widehat{Q} = \{\{u, v\} \mid \text{there is a } u - v \text{ path } P \text{ in } Q\}$. Fix any $\ell \in [q]$ and define C_ℓ to be the set of connected components when we restrict C to H_ℓ . That is, each element in C_ℓ is a path (maybe of length 0) or C itself (in that case $C_\ell = \{C\}$). We also use C_ℓ to denote the subgraph of C restricted to H_ℓ . See Fig. 2 for an illustration. Notice that $\bigcup_{F \in \widehat{C}_\ell} F$ is the set of vertices of degree 0 or 1 in $H_\ell(C)$ and $\bigcup_{F \in \widehat{C}_\ell} F \subseteq X_\ell$. Since X_ℓ is a union of vertices from at most $7\sqrt{k}$ many cells of f , C is a simplified cycle, and by property 2 of the clique-grid graph, we have that the cardinality of $\bigcup_{F \in \widehat{C}_\ell} F$

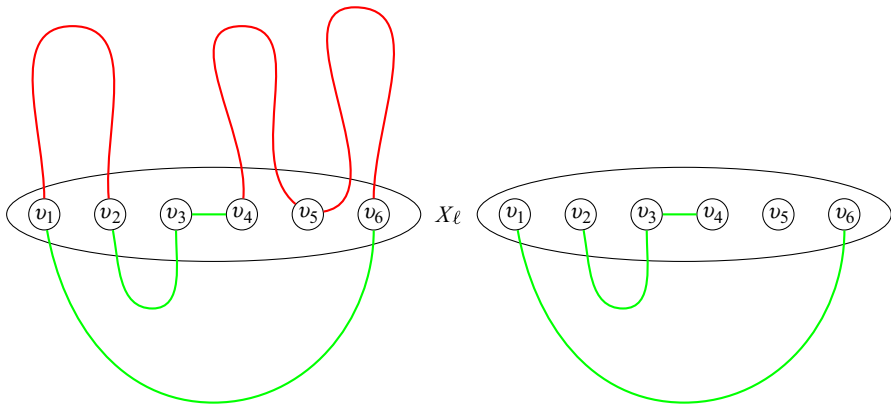


Fig. 2 Illustration of cycle C intersecting X_ℓ . The figure in the right side represents C_ℓ . Here the set \widehat{C}_ℓ is $\{\{v_1, v_6\}, \{v_2, v_4\}, \{v_5\}\}$

is at most $5 \cdot 24 \cdot 7\sqrt{k} = 840\sqrt{k}$. In our DP algorithm we will have a state indexed by $(\ell, \widehat{C}_\ell, |E(C_\ell)|)$ of Boolean value, which will be set to 1. Formally, for any $\ell \in [q]$, $k' \in [k]$ and a family \mathcal{Z} of vertex-disjoint sets of size at most 2 of X_ℓ with the property that the cardinality of $\bigcup_{Z \in \mathcal{Z}} Z$ is at most $840\sqrt{k}$, we will have a DP table entry $\mathcal{A}[\ell, \mathcal{Z}, k']$. For each $\ell \in [q]$, we maintain the following correctness invariants.

Correctness Invariants:

- (i) For every $C \in \mathcal{C}$, $\mathcal{A}[\ell, \widehat{C}_\ell, |E(C_\ell)|] = 1$,
- (ii) for any family \mathcal{Z} of vertex-disjoint sets of size at most 2 of X_ℓ with $0 < |\bigcup_{Z \in \mathcal{Z}} Z| \leq 840\sqrt{k}$, $k' \in [k]$, and $\mathcal{A}[\ell, \mathcal{Z}, k'] = 1$, there is a set Q of $|\mathcal{Z}|$ vertex-disjoint paths in H_ℓ where the endpoints of each path are specified by a set in \mathcal{Z} and $|E(Q)| = k'$, and
- (iii) if $\mathcal{A}[\ell, \emptyset, k] = 1$, then there is a cycle of length k in H_ℓ .

The correctness of the algorithm will follow from the correctness invariants for $\ell = q$. In what follows we explain how to compute $\mathcal{A}[\ell, \mathcal{Z}, k']$ for every $\ell \in [q]$, $k' \in [k]$ and family \mathcal{Z} of vertex-disjoint sets of size at most 2 of X_ℓ with $|\bigcup_{Z \in \mathcal{Z}} Z| \leq 840\sqrt{k}$, the running time to compute it from the previously computed DP table entries, and prove the correctness invariants. While proving the correctness invariants for ℓ , we assume that the correctness invariants hold for $\ell - 1$. When $\ell = 1$, $X_1 = \emptyset$ and the DP table entries are defined as follows:

$$\mathcal{A}[1, \emptyset, k'] = \begin{cases} 1 & \text{if } k' = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Since $H_1 = (\emptyset, \emptyset)$, the correctness invariants follow. The values $\mathcal{A}[1, *, *]$ can be computed in $\mathcal{O}(1)$ time. Now we move to the case where $\ell > 1$.

Case 1: X_ℓ is a forget bag. Fix a family \mathcal{Z} of vertex-disjoint sets of size at most 2 of X_ℓ such that $|\bigcup_{Z \in \mathcal{Z}} Z|$ is at most $840\sqrt{k}$ and $k' \in [k]$.

$$A[\ell, \mathcal{Z}, k'] = A[\ell - 1, \mathcal{Z}, k']. \tag{2}$$

Clearly $A[\ell, \mathcal{Z}, k']$ can be computed in $\mathcal{O}(1)$ time using the previously computed DP table entries. In this case the correctness invariants follow from the fact that $H_\ell = H_{\ell-1}$ and $X_\ell \subseteq X_{\ell-1}$.

Case 2: X_ℓ is an introduce bag. That is $X_\ell = X_{\ell-1} \cup f^{-1}(L)$ for some cell L . Fix a family \mathcal{Z} of vertex-disjoint sets of size at most 2 of X_ℓ such that $|\bigcup_{Z \in \mathcal{Z}} Z|$ is at most $840\sqrt{k}$ and $k' \in [k]$. Define $\mathcal{Q}_\ell(r)$ to be the family of sets of vertex-disjoint paths in $H[X_\ell \setminus X_{\ell-1}] = H[f^{-1}(L)]$ as follows. Each set of paths $Q \in \mathcal{Q}_\ell(r)$ has at most 120 endpoints and $|E(Q)| = r$. Any set Q that satisfies the above properties will be in $\mathcal{Q}_\ell(r)$. Recall that $\widehat{Q} = \{\{u, v\} \mid \text{there is a } u - v \text{ path in } Q\}$. Define $\widehat{\mathcal{Q}}_\ell(r) = \{\widehat{Q} : Q \in \mathcal{Q}_\ell(r)\}$.

Claim 2 $|\widehat{\mathcal{Q}}_\ell(r)| = k^{\mathcal{O}(1)}$ and $\widehat{\mathcal{Q}}_\ell(r)$ can be enumerated in time $k^{\mathcal{O}(1)}$.

Proof We know that $H[f^{-1}(L)]$ is a complete graph. For any family \mathcal{W} of vertex-disjoint sets of size at most 2 of $f^{-1}(L)$, one can easily get $|\mathcal{W}|$ paths with endpoints being the one specified by the sets in \mathcal{W} , where each path is of length 0 or 1. To get $|\mathcal{W}|$ vertex-disjoint paths with endpoints being the one specified by the sets in \mathcal{W} with total number of edges r , there should be enough vertices in $f^{-1}(L) \setminus (\bigcup_{W \in \mathcal{W}} W)$. That is, when one of the following conditions holds: (i) either $|\{W \in \mathcal{W} : |W| = 2\}| = r$, or (ii) $1 \leq |\{W \in \mathcal{W} : |W| = 2\}| < r$ and $|\{W \in \mathcal{W} : |W| = 2\}| + |f^{-1}(L) \setminus \bigcup_{W \in \mathcal{W}} W| \geq r$. Moreover, since the number of vertices in the union of endpoints of paths in a set $Q \in \mathcal{Q}_\ell(r)$ is at most 120 and $|V(H)| = k^{\mathcal{O}(1)}$, we can enumerate $\widehat{\mathcal{Q}}_\ell(r)$ in time $k^{\mathcal{O}(1)}$. \square

For any family \mathcal{Y} of sets of size at most 2 in X_ℓ , we say that \mathcal{Y} forms a family of paths (respectively, a cycle) in $K[X_\ell]$, if the graph $(\bigcup_{Y \in \mathcal{Y}} Y, \{e \in \mathcal{Y} : |e| = 2\})$ forms a family of paths (respectively, a cycle). Now we move to the computation of $A[\ell, \mathcal{Z}, k']$. Consider the case when $\mathcal{Z} \neq \emptyset$.

$$\text{If } \mathcal{Z} \in \widehat{\mathcal{Q}}_\ell(k'), \text{ then we set } A[\ell, \mathcal{Z}, k'] = 1. \tag{3}$$

Otherwise,

$$A[\ell, \mathcal{Z}, k'] = \max \left\{ A[\ell - 1, \mathcal{Z}', k''] : \mathcal{Z}' \neq \emptyset \text{ and there exist } r \in \mathbb{N}, \widehat{\mathcal{Q}} \in \widehat{\mathcal{Q}}_\ell(r), \right. \\ E' \subseteq E \left(\bigcup_{Q \in \widehat{\mathcal{Q}}} Q, \bigcup_{Z \in \mathcal{Z}'} Z \right) \text{ such that } |E'| \leq 120, k' = k'' + r + |E'|, \\ \left. \mathcal{Z}' \cup \widehat{\mathcal{Q}} \cup E' \text{ forms a set } \mathcal{R} \text{ of paths in } K[X_\ell], \text{ and } \widehat{\mathcal{R}} = \mathcal{Z} \right\}. \tag{4}$$

Now consider the case when $\mathcal{Z} = \emptyset$.

$$\text{If } \mathcal{A}[\ell - 1, \emptyset, k'] = 1, \text{ then we set } \mathcal{A}[\ell, \emptyset, k'] = 1. \tag{5}$$

Otherwise

$$\begin{aligned} \mathcal{A}[\ell, \emptyset, k'] = \max \left\{ \mathcal{A}[\ell - 1, \mathcal{Z}', k''] : \text{there exist } r \in \mathbb{N}, \widehat{Q} \in \widehat{Q}_\ell(r), \right. \\ \left. E' \subseteq E \left(\bigcup_{Q \in \widehat{Q}} Q, \bigcup_{Z \in \mathcal{Z}'} Z \right) \text{ such that } k' = k'' + r + |E'|, \right. \\ \left. |E'| \leq 120, \text{ and } \mathcal{Z}' \cup \widehat{Q} \cup E' \text{ forms a cycle in } K[X_\ell] \right\}. \tag{6} \end{aligned}$$

Notice that in the above computation [(4) and (6)] the number of potential choices for \mathcal{Z}' is bounded by $2^{\mathcal{O}(\sqrt{k} \log k)}$. By Claim 2 we know that the cardinality of $\widehat{Q}_\ell(r)$ is at most $k^{\mathcal{O}(1)}$ and it can be enumerated in time $k^{\mathcal{O}(1)}$. Since $|X_\ell| = k^{\mathcal{O}(1)}$, the number of choices for E' in the above computation is bounded by $k^{\mathcal{O}(1)}$. This implies that we can compute $\mathcal{A}[\ell, \mathcal{Z}, k']$ using previously computed DP table entries in time $2^{\mathcal{O}(\sqrt{k} \log k)}$.

Before proving the correctness invariants, we state the following simple claim.

Claim 3 For any $\ell \in [q]$, $\mathcal{A}[\ell, \emptyset, 0] = 1$.

Proof When $\ell = 1$, the claim follows from (1). For $\ell = 2, \dots, q$, the claim follows from (2) or (5). □

Now we prove the correctness invariants. Let $C \in \mathcal{C}$. We partition the set of edges $E(C_\ell)$ into $E_1 \uplus E_2 \uplus E'$ where $E_1 = E(C_\ell) \cap E(H_{\ell-1})$, $E_2 = E(C_\ell) \cap E(X_\ell \setminus X_{\ell-1})$ and $E' = E(C_\ell) \cap E(X_{\ell-1}, X_\ell \setminus X_{\ell-1})$. We have two cases based on whether $\widehat{C}_\ell = \emptyset$ or not.

Suppose $\widehat{C}_\ell \neq \emptyset$. If C_ℓ is a subgraph of $H[X_\ell \setminus X_{\ell-1}]$, then $\widehat{C}_\ell \in \widehat{Q}_\ell(|E(C_\ell)|)$. Then by (3), we have that $\mathcal{A}[\ell, \widehat{C}_\ell, |E(C_\ell)|] = 1$. So now we have that C_ℓ is not a subgraph of $H[X_\ell \setminus X_{\ell-1}]$. This implies that either $E_1 \neq \emptyset$ or $E' \neq \emptyset$. In either case, we have that $\widehat{C}_{\ell-1} \neq \emptyset$. Since $X_{\ell-1}$ is a union of vertices from $7\sqrt{k}$ cells, $C \in \mathcal{C}$ and by property 2 of clique-grid graphs, we have that $|E(X_{\ell-1}, V(H) \setminus X_{\ell-1})| \leq 5 \cdot 24 \cdot 7\sqrt{k} = 840\sqrt{k}$. This implies that $|\bigcup_{D \in \widehat{C}_{\ell-1}} D| \leq 840\sqrt{k}$. By statement (ii) of the correctness invariant for $\ell - 1$, we have that (a) $\mathcal{A}[\ell - 1, \widehat{C}_{\ell-1}, |E(C_{\ell-1})|] = 1$. Consider the graph $H[E_2]$. The graph $H[E_2]$ is a collection Q of paths in $H[X_\ell \setminus X_{\ell-1}] = K[f^{-1}(L)]$. Since $C \in \mathcal{C}$, the number of edges of C with one endpoint in $f^{-1}(L)$ and other in $X_{\ell-1}$ is at most $5 \cdot 24 = 120$. This implies that (b) $|E'| \leq 120$ and (c) $\widehat{Q} \in \widehat{Q}_\ell(r)$, where $r = |E(Q)|$. By facts (a), (b) and (c), using (4), we get $\mathcal{A}[\ell, \widehat{C}_\ell, |E(C_\ell)|] = 1$.

Now consider the case when $\widehat{C}_\ell = \emptyset$. In this case either $E(C) \cap E(H_\ell) = \emptyset$ or $E(C) \subseteq E(H_\ell)$. If $E(C) \cap E(H_\ell) = \emptyset$, then $|E(C_\ell)| = 0$ and hence $\mathcal{A}[\ell, \widehat{C}_\ell, |E(C_\ell)|] = \mathcal{A}[\ell, \emptyset, 0] = 1$, by Claim 3. Now we have $\widehat{C}_\ell = \emptyset$ and $E(C) \subseteq E(H_\ell)$. This implies that $|E(C_\ell)| = k$. If $E(C_\ell) = E(C_{\ell-1})$, then

$\mathcal{A}[\ell - 1, \widehat{C}_\ell, k] = \mathcal{A}[\ell - 1, \emptyset, k] = 1$, by statement (i) of the correctness invariants for $\ell - 1$. Then, by (5), $\mathcal{A}[\ell, \emptyset, k] = 1$. So, now we assume that $E(C_\ell) \neq E(C_{\ell-1})$. This implies that either $E_2 \neq \emptyset$ or $E' \neq \emptyset$. Since $|f^{-1}(L)| < k$, $C_\ell = \{C\}$ and $E(C_\ell) \neq E(C_{\ell-1})$, we have that $\widehat{C}_{\ell-1} \neq \emptyset$. The graph $H[E_2]$ is a collection Q of paths in $H[X_\ell \setminus X_{\ell-1}] = K[f^{-1}(L)]$. As like before, we can bound (d) $|\bigcup_{D \in \widehat{C}_{\ell-1}} D| \leq 840\sqrt{k}$, (e) $|E'| \leq 120$ and (g) $\widehat{Q} \in \widehat{Q}_\ell(r)$, where $r = |E(Q)|$. By (d) and statement (ii) of the correctness invariants for $\ell - 1$, we get (h) $\mathcal{A}[\ell - 1, \widehat{C}_{\ell-1}, |E(C_{\ell-1})|] = 1$. By facts (h), (e) and (d), using (6), we get $\mathcal{A}[\ell, \widehat{C}_\ell, |E(C_\ell)|] = \mathcal{A}[\ell, \emptyset, k] = 1$. This completes the proof of statement (i).

Now we need to prove statement (ii) of the correctness invariants. Let \mathcal{Z} be a family of vertex-disjoint sets of size at most 2 of X_ℓ with $0 < |\bigcup_{Z \in \mathcal{Z}} Z| \leq 840\sqrt{k}$ and $k' \in [k]$. Notice that $\mathcal{Z} \neq \emptyset$. Suppose in the above computation we set $\mathcal{A}[\ell, \mathcal{Z}, k'] = 1$. Either $\mathcal{A}[\ell, \mathcal{Z}, k']$ is set to 1 because of (3) or because of (4). If $\mathcal{A}[\ell, \mathcal{Z}, k']$ is set to 1 because of (3), then we know that $\mathcal{Z} \in \widehat{Q}_\ell(k')$. By the definition of $\widehat{Q}_\ell(k')$, we get that there is a set R of vertex-disjoint paths in $H[X_\ell \setminus X_{\ell-1}]$, hence in $H[X_\ell]$ and $\widehat{R} = \mathcal{Z}$. So, now we assume that $\mathcal{A}[\ell, \mathcal{Z}, k']$ is set to 1 because of (4). This implies that there exist $k'', r \in \mathbb{N}$, a family \mathcal{Z}' of vertex-disjoint sets of size at most 2 of $X_{\ell-1}$, $\widehat{Q} \in \widehat{Q}_\ell(r)$, and $E' \subseteq E(\bigcup_{Q \in \widehat{Q}} Q, \bigcup_{Z \in \widehat{\mathcal{Z}}'} Z)$ such that $\mathcal{A}[\ell - 1, \mathcal{Z}', k''] = 1$, $|E'| \leq 120$, $k' = k'' + r + |E'|$, $\mathcal{Z}' \cup \widehat{Q} \cup E'$ forms a set of paths \mathcal{R} in $K[X_\ell]$ with $\widehat{\mathcal{R}} = \mathcal{Z}$. Since $\mathcal{A}[\ell - 1, \mathcal{Z}', k''] = 1$, by statement (ii) of the correctness invariants for $\ell - 1$, we have that there is a set \mathcal{Y} of $|\mathcal{Z}'|$ vertex-disjoint paths in $H_{\ell-1}$ where the endpoints of each path are specified by a set in \mathcal{Z}' and $|E(\mathcal{Y})| = k''$. Let Q be the set of paths in $Q_\ell(r)$ corresponding to the set \widehat{Q} . Thus by replacing each edge of \mathcal{Z}' in $\mathcal{Z}' \cup \widehat{Q} \cup E'$ by the corresponding path in \mathcal{Y} and each edge of \widehat{Q} by a corresponding path in Q , we can get a set \mathcal{W} of vertex-disjoint paths in H_ℓ , because the internal vertices of paths in \mathcal{Y} are disjoint from $(X_\ell \setminus X_{\ell-1}) \cup \bigcup_{Z \in \mathcal{Z}} Z$ and the internal vertices of paths in Q are disjoint from $X_{\ell-1} \cup \bigcup_{Z \in \mathcal{Z}} Z$. Moreover, $\widehat{\mathcal{W}} = \mathcal{Z}$ and $|E(\mathcal{W})| = |E(\mathcal{Y})| + |E(Q)| + |E'| = k'' + r + |E'| = k'$. This completes the proof of statement (ii) in the correctness invariants.

Now we prove statement (iii) in the correctness invariants. Suppose we set $\mathcal{A}[\ell, \emptyset, k] = 1$. Then either $\mathcal{A}[\ell, \emptyset, k]$ is set to 1 because of (5) or because of (6). If $\mathcal{A}[\ell, \emptyset, k]$ is set to 1 because of (5), then we know that $\mathcal{A}[\ell - 1, \emptyset, k] = 1$, then by statement (iii) of the correctness invariants for $\ell - 1$, we have that there is a cycle of length k in $H_{\ell-1}$ and hence in H_ℓ . Suppose $\mathcal{A}[\ell, \emptyset, k]$ is set to 1 because of (6). Now the arguments for the proof is similar to that of the case for statement (ii), when $\mathcal{A}[\ell, \mathcal{Z}, k']$ is set to 1 because of (4). This completes the proof of statement (iii) in the correctness invariants.

Algorithm \mathcal{A} outputs YES if $\mathcal{A}[q, \emptyset, k] = 1$ and a outputs No otherwise. The correctness of the algorithm follows from the correctness invariants. Now we analyse the total running time. Notice that $|V(H)| = k^{O(1)}$ and number of DP table entries is bounded by $2^{O(\sqrt{k} \log k)}$. Each DP table entry can be computed in time $2^{O(\sqrt{k} \log k)}$ using the previously stored values in the DP table. Hence the total running time of the algorithm is $2^{O(\sqrt{k} \log k)}$. □

Lemmata 6.2, 6.4 and 6.5 imply the following lemma.

Lemma 6.6 *Let $(G, f : V(G) \rightarrow [2k] \times [2k], k)$ be an instance of EXACT k -CYCLE, where G is a clique-grid graph with representation f , $|f^{-1}(L)| < k$ for all $L \in [2k] \times [2k]$. There is an algorithm which given $(G, f : V(G) \rightarrow [2k] \times [2k], k)$, runs in time $2^{\mathcal{O}(\sqrt{k} \log k)}$, and decides whether G has a cycle of length k or not.*

Corollary 5.6 and Lemma 6.6 imply Theorem 6.1. We can design a similar algorithm to solve LONGEST PATH in time $2^{\mathcal{O}(\sqrt{k} \log k)} n^{\mathcal{O}(1)}$.

7 Longest Cycle

In this section, we show that LONGEST CYCLE admits a subexponential-time parameterized algorithm. More precisely, we prove the following.

Theorem 7.1 LONGEST CYCLE on unit disk/square graphs can be solved in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$.

To prove Theorem 7.1 we first apply Corollary 5.6 and then solve the problem on each output instance $(\widehat{G}, \widehat{f}, k)$ where \widehat{f} is a representation of \widehat{G} . We start with the definition of contractible edges and state some simple results to prove Theorem 7.1.

Definition 7.2 Let G be a clique-grid graph with a representation f . A pair (u, v) of distinct vertices $u, v \in V(G)$ is *contractible* if $f(u) = f(v)$.

Note that if (u, v) is a contractible pair, then by Condition 1 in Definition 3.1, it holds that $\{u, v\} \in E(G)$. Now, given a contractible pair (u, v) , denote $e = \{u, v\}$, and define the function $f_{/e} : V(G/e) \rightarrow [t] \times [t']$ as follows. For all $w \in V(G) \setminus \{u, v\}$, define $f_{/e}(w) = f(w)$. Moreover, define $f_{/e}(x_{\{u,v\}}) = f(u)$ (Recall that $\{x_{\{u,v\}}\} = V(G/e) \setminus V(G)$). By Definitions 3.1 and 4.6, we immediately have the following.

Observation 7.3 G/e is a clique-grid graph and the function $f_{/e}$ is a representation of G/e . Furthermore, G and G/e have the same cell graph.

Now, we verify that in case there exists a cycle on at least $2k$ vertices, the operation that contracts an edge also preserves the answer YES for LONGEST CYCLE.

Lemma 7.4 Let (G, f, k) be an instance of LONGEST CYCLE on clique-grid graphs such that G contains a cycle C on at least $2k$ vertices. Then, $(G/e, f_{/e}, k)$ is a YES-instance.

Proof Denote $e = \{u, v\}$. In case $V(C) \cap \{u, v\} = \emptyset$, then C is also a cycle in G/e , and in case $|V(C) \cap \{u, v\}| = 1$, then by replacing the vertex in $V(C) \cap \{u, v\}$ by $x_{\{u,v\}}$ in C , we obtain a cycle of the same length as C in G/e . In both of these cases, the proof is complete, and thus we next suppose that $\{u, v\} \subseteq V(C)$.

Let us denote $C = v_1 - v_2 - v_3 - \dots - v_\ell - v_1$, where $v_1 = u$ and $v_i = v$ for some $i \in [\ell] \setminus \{1\}$. Note that $\ell \geq 2k$. Without loss of generality, assume that $i - 2 \geq \ell - i$ (else we replace each v_j , except for v_1 , by $v_{\ell+2-j}$, and obtain a cycle where this property holds). Now, note that $C' = x_{\{u,v\}} - v_2 - \dots - v_{i-1} - x_{\{u,v\}}$ is a cycle in G/e . Moreover, since $i - 2 \geq \ell - i$, it holds that $|V(C')| = i - 1 \geq \frac{\ell}{2} \geq k$. Thus, $(G/e, f_{/e}, k)$ is a YES-instance. □

Before we present our algorithm, we need two additional propositions, handling the extreme cases where we either discover that our input graph contains a large grid or, after a series of operations that contracted edges in G , we ended up with a graph isomorphic to the cell graph of G . For the first case, we need the following result (see also [13,16]).

Observation 7.5 *Let (G, k) be an instance of LONGEST CYCLE on general graphs. If G contains a $\lceil\sqrt{k}\rceil \times \lceil\sqrt{k}\rceil$ grid as a minor, then (G, k) is a YES-instance.*

For the second case, we need the following result.

Observation 7.6 *Let (G, f, k) be an instance of LONGEST CYCLE on clique-grid graphs. Then, it can be determined in time $2^{\mathcal{O}(tw(\text{cell}(G)))} \cdot n^{\mathcal{O}(1)}$ whether $\text{cell}(G)$ contains a cycle on at least k vertices, in which case (G, f, k) is a YES-instance.*

Proof There is an algorithm for LONGEST CYCLE running in time $2^{\mathcal{O}(tw(G))} \cdot n^{\mathcal{O}(1)}$ [6]. Since $\text{cell}(G)$ is a minor of G , the proof of the observation follows. \square

We are now ready to present our algorithm. The proof of correctness and analysis of running times are integrated in the description of the algorithm.

Proof of Theorem 7.1 Let (G, O, k) be an instance of LONGEST CYCLE on unit disk/square graphs. By using Corollary 5.6, in polynomial time we get a set \mathcal{I} of instances of LONGEST CYCLE on clique-grid graphs such that (G, O, k) is a YES-instance if and only if at least one instance in \mathcal{I} is a YES-instance, and for all $(\widehat{G}, \widehat{f}, \widehat{k}) \in \mathcal{I}$, it holds that \widehat{G} is either an induced subgraph of G or $K_{\widehat{k}}, \widehat{t}, \widehat{t}' \leq 2\widehat{k}$, $|\widehat{f}^{-1}(L)| < \widehat{k}$ for any cell $L \in [\widehat{t}] \times [\widehat{t}']$ and $\widehat{k} = k$.

Now we fix an instance $(\widehat{G}, \widehat{f}, k) \in \mathcal{I}$. In the rest of the proof we explain how to solve the instance $(\widehat{G}, \widehat{f}, k)$. Next, by using Lemma 4.9 with the parameter $\ell = 100 \cdot 599^3 \cdot \lceil\sqrt{k}\rceil$, we either correctly conclude that \widehat{G} contains a $\lceil\sqrt{k}\rceil \times \lceil\sqrt{k}\rceil$ grid as a minor, or compute a tree decomposition of $\text{cell}(\widehat{G})$ of width at most $500 \cdot 599^3 \cdot \lceil\sqrt{k}\rceil = \mathcal{O}(\sqrt{k})$. In the first case, by Observation 7.5, we are done. In the latter case, by using Observation 7.6, we determine in time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ whether $\text{cell}(\widehat{G})$ contains a cycle on at least k vertices, where if the answer is positive, then we are done. Thus, we next suppose that $\text{cell}(\widehat{G})$ does not contain a cycle on at least k vertices.

Now, as long as there exists a contractible pair (u, v) , we perform the following operation. First, by using Lemma 6.6, we determine in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ whether \widehat{G} contains a cycle whose number of vertices is between k and $2k$. If the answer is positive, then we are done (our final answer is YES). If the answer is negative, then we contract the edge $\{u, v\}$. We know that (i) either \widehat{G} has a cycle of length at least $2k$ or it has no cycle of length at least k . That is, if \widehat{G} has a cycle C of length at least k , then by statement (i), $|V(C)| \geq 2k$, and hence by Lemma 7.4, we obtain an instance (after contracting $\{u, v\}$) that has a cycle of length at least k . If \widehat{G} has no cycle of length at least k , then clearly the instance (after contracting $\{u, v\}$) has no cycle of length at least k . Note that the loop described in this paragraph can have at most $\mathcal{O}(n^2)$ iterations, and therefore its total running time is bounded by $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$. Once there does not exist a contractible pair (u, v) , as we have only modified the graph

by contracting edges, one at a time, between contractible pairs, we are left with a graph that is isomorphic to the cell graph of our original input graph. We have already correctly concluded that this graph does not contain a cycle on at least k vertices. Thus, at this point, we correctly conclude that $(\widehat{G}, \widehat{f}, k)$ is a NO-instance. \square

8 Feedback Vertex Set

In this section, we show that FEEDBACK VERTEX SET admits a subexponential-time parameterized algorithm. More precisely, we prove the following.

Theorem 8.1 FEEDBACK VERTEX SET on unit disk/square graphs can be solved in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$.

First, we prove that there is a $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ time algorithm which either concludes that there is no feedback vertex set of size k or outputs an $\mathcal{O}(\sqrt{k})$ -NCTD of the input graph.

Lemma 8.2 Let (G, O, k) be an instance of FEEDBACK VERTEX SET on unit disk/square graphs. Then, in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$, one can either solve (G, O, k) or obtain an equivalent instance (G, f, k) of FEEDBACK VERTEX SET on clique-grid graphs together with an $\mathcal{O}(\sqrt{k})$ -NCTD of G such that $|f^{-1}(L)| \leq k + 2$ for any cell L .

Proof First, by using Lemmata 3.2 or 3.3, we obtain a representation f of G . Notice that if there is a cell L of f , such that $|f^{-1}(L)| \geq k + 3$, then there is no feedback vertex set of size at most k in G , because $f^{-1}(L)$ is a clique of size at least $k + 3$ in G . Now, by using Corollary 4.10 with $\ell = 200 \cdot 599^3 \cdot (\lceil \sqrt{k} \rceil + 1) = \mathcal{O}(\sqrt{k})$, we either correctly conclude that G contains a $2(\lceil \sqrt{k} \rceil + 1) \times 2(\lceil \sqrt{k} \rceil + 1)$ grid as a minor, or compute an $\mathcal{O}(\sqrt{k})$ -NCTD of G . If there is a $2(\lceil \sqrt{k} \rceil + 1) \times 2(\lceil \sqrt{k} \rceil + 1)$ grid as a minor, then there are more than k vertex-disjoint cycles in G and hence (G, O, k) is a NO-instance. \square

Because of Lemma 8.2, to prove Theorem 8.1, we can focus on FEEDBACK VERTEX SET on clique-grid graphs, where the input also contains an $\mathcal{O}(\sqrt{k})$ -NCTD. That is, the input of FEEDBACK VERTEX SET on clique-grid graphs is a tuple (G, f, k, \mathcal{T}) where G is a clique-grid graph with representation f , $|f^{-1}(L)| \leq k + 2$ for all cells L of f and $\mathcal{T} = (T, \beta)$ is an $\mathcal{O}(\sqrt{k})$ -NCTD of G . The following observation follows from the fact that $\mathcal{T} = (T, \beta)$ is an $\mathcal{O}(\sqrt{k})$ -NCTD and $|f^{-1}(L)| \leq k + 2$ for any cell L of f .

Observation 8.3 For any $v \in V(T)$, $|\beta(v)| = \mathcal{O}(k^{1.5})$.

Notice that G has a feedback vertex set of size at most k if and only if there is a vertex subset $F \subseteq V(G)$ of cardinality at least $|V(G)| - k$ such that $G[F]$ is a forest. Hence, instead of stating the problem as finding a k sized feedback vertex set in G , we can state it as finding an induced subgraph H of G with maximum number of vertices such that H is a forest.

MAX INDUCED FOREST (MIF)

Parameter: k

Input: A clique-grid graph G with representation f and an integer k such that \mathcal{T} is a $c\sqrt{k}$ -NCTD of G and for any cell L in f , $|f^{-1}(L)| \leq k + 2$, where c is a constant.

Question: Is there subset $W \subseteq V(G)$ such that $G[W]$ is a forest and $|W| \geq |V(G)| - k$?

Observation 8.4 *Let (G, f, k, \mathcal{T}) be an instance of MIF. Then (G, f, k, \mathcal{T}) is a YES-instance of MIF if and only if (G, f, k, \mathcal{T}) is a YES-instance of FEEDBACK VERTEX SET.*

By Lemmas 8.2 and 8.4, to prove Theorem 8.1, it is sufficient that we prove the following result (which is the focus of the rest of this section).

Lemma 8.5 *MIF on clique-grid graphs can be solved in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$.*

Proof (Sketch) We explain a DP algorithm which given as input (G, f, k, \mathcal{T}) where G is a clique-grid graph with representation f , $\mathcal{T} = (T, \beta)$ is a $c\sqrt{k}$ -NCTD, c is a constant and $|f^{-1}(L)| \leq k + 2$ for any cell L of f and outputs YES if there is an induced forest with at least $|V(G)| - k$ vertices and outputs NO otherwise. Here we use the term solution for a vertex subset $S \subseteq V(G)$ with the property that $G[S]$ is a forest. First notice that any solution S contains at most 2 vertices from $f^{-1}(L)$ for any cell L of f . Now, the following claim follows from the fact that \mathcal{T} is a $c\sqrt{k}$ -NCTD and any solution contains at most 2 vertices from $f^{-1}(L)$ for any cell L .

Claim 4 For any $v \in V(T)$ and any solution S , $|S \cap \beta(v)| \leq 2c\sqrt{k}$.

We first briefly explain what the table entries are in a standard DP algorithm for our problem on graphs of bounded treewidth [13]. Then we explain that in fact many of the entries we compute in the standard DP table is redundant in our case, because of Observation 8.3 and Claim 4. That is, Observation 8.3 and Claim 4 show that only $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot |V(G)|^{\mathcal{O}(1)}$ many states in the DP table are relevant in our case. Recall that for any $v \in V(T)$, $\gamma(v)$ denotes the union of the bags of v and its descendants. The standard DP table for our problem will have an entry indexed by $(v, U, \{U_1, U_2, \dots, U_\ell\})$ where $v \in V(T)$, $U \subseteq \beta(v)$ and $U_1 \uplus U_2 \uplus \dots \uplus U_\ell = U$. The table entry $\mathcal{A}[v, U, \{U_1, U_2, \dots, U_\ell\}]$ stores the following information: the maximum cardinality of a vertex subset $W \subseteq G[\gamma(v)]$ such that $W \cap \beta(v) = U$, $G[W]$ is a forest with a set of connected components \mathcal{C} and for any $C \in \mathcal{C}$, either $V(C) \cap \beta(v) = \emptyset$ or $V(C) \cap \beta(v) = U_i$ for some $i \in [\ell]$. Notice that the total number of DP table entries is bounded by $tw^{\mathcal{O}(tw)} \cdot |V(G)|^{\mathcal{O}(1)}$ where tw is the width of the tree decomposition \mathcal{T} . One can easily show that the computation of the DP table at a node can be done in time polynomial in the size of the tables of its children.

By Observation 8.3 and Claim 4, we know that for any bag $\beta(v)$ in \mathcal{T} , the potential number of subsets of $\beta(v)$ which can be part of any solution is at most $2^{\mathcal{O}(\sqrt{k} \log k)}$. This implies that we only need to compute the DP table entries for indices $(v, U, \{U_1, U_2, \dots, U_\ell\})$ where $v \in V(T)$, $U \subseteq \beta(v)$ and $|U| \leq 2c\sqrt{k}$. Thus, the size of the DP table, and hence the time to compute it takes $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ time. This concludes the description. □

9 Cycle Packing

In this section, we show that CYCLE PACKING admits a subexponential-time parameterized algorithm. More precisely, we prove the following.

Theorem 9.1 *CYCLE PACKING on unit disk/square graphs can be solved in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$.*

First, we prove that there is a $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ time algorithm which either concludes that there are k vertex-disjoint cycles or outputs an $\mathcal{O}(\sqrt{k})$ -NCTD of the input graph.

Lemma 9.2 *Let (G, O, k) be an instance of CYCLE PACKING on unit disk/square graphs. Then, in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$, one can either solve (G, O, k) or obtain an equivalent instance (G, f, k) of CYCLE PACKING on clique-grid graphs together with an $\mathcal{O}(\sqrt{k})$ -NCTD of G such that $|f^{-1}(L)| \leq 3k$ for all cells L of f .*

Proof First, by using Lemmata 3.2 or 3.3, we obtain a representation f of G . If there exists a cell L of f such that $|f^{-1}(L)| \geq 3k$, then by Condition 1 in Definition 3.1, $G[f^{-1}(L)]$ is a clique on at least $3k$ vertices and thus it contains k pairwise vertex-disjoint cycles (triangles). Now, by using Corollary 4.10 with $\ell = 200 \cdot 599^3 \cdot \lceil \sqrt{k} \rceil = \mathcal{O}(\sqrt{k})$, we either correctly conclude that G contains a $2\lceil \sqrt{k} \rceil \times 2\lceil \sqrt{k} \rceil$ grid as a minor, or compute an $\mathcal{O}(\sqrt{k})$ -NCTD of G . If G contains $2\lceil \sqrt{k} \rceil \times 2\lceil \sqrt{k} \rceil$ grid as a minor, then G has k vertex-disjoint cycles. \square

By Lemma 9.2 to prove Theorem 9.1, it is sufficient that we prove the following result (which is the focus of the rest of this section).

Lemma 9.3 *CYCLE PACKING on clique-grid graphs can be solved in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$, assuming that the input includes an $\mathcal{O}(\sqrt{k})$ -NCTD of G , and that for every cell L of f , it holds that $|f^{-1}(L)| \leq 3k$.*

Let $(G, f : V(G) \rightarrow [t] \times [t'], k)$ denote the input instance of CYCLE PACKING, and let $\mathcal{T} = (T, \beta)$ denote our $\mathcal{O}(\sqrt{k})$ -NCTD of G . For simplicity, from now on, let $\mathcal{L} = [t] \times [t']$. Note that since \mathcal{T} is an $\mathcal{O}(\sqrt{k})$ -NCTD, and since for every $L \in \mathcal{L}$, it holds that $|f^{-1}(L)| \leq 3k$, we also have the following.

Observation 9.4 *For all $v \in V(T)$, it holds that $|\beta(v)| = \mathcal{O}(k^{1.5})$.*

We proceed by considering the “interaction” between cells in the context of the manner in which cycles in a solution cross their boundaries. To be precise, by Definition 3.1, we first observe the following.

Observation 9.5 *Let C be an induced cycle in G . Then, there does not exist a cell $L \in \mathcal{L}$ and two distinct vertices $u, v \in V(C) \cap f^{-1}(L)$ such that $\{u, v\} \notin E(C)$. In particular, for every cell $L \in \mathcal{L}$, exactly one of the following conditions holds:*

1. $V(C) \subseteq f^{-1}(L)$.
2. $|V(C) \cap f^{-1}(L)| = 1$.

3. $|V(C) \cap f^{-1}(L)| = 2$ and the two vertices in $V(C) \cap f^{-1}(L)$ are neighbors in C .

That is, any induced cycle C in G can have at most 2 vertices in a cell and if there are two vertices in a cell, then they are adjacent in C . Hence the interaction of an induced cycle from a cell to the rest of the cells is at most 2 (in other words the number of edges of C with one endpoint in a cell L and other endpoint is not in L is at most 2). Next, note that given a set \mathcal{C} of pairwise vertex-disjoint cycles and a cycle $C \in \mathcal{C}$ that is not an induced cycle in G , by replacing C in \mathcal{C} by an induced cycle in $G[V(C)]$, we obtain another set of pairwise vertex-disjoint cycles. Thus, we have the following.

Observation 9.6 *If (G, f, k) is a YES-instance, then G contains a set \mathcal{C} of k pairwise-disjoint induced cycles.*

Definition 9.7 Given two distinct cells $L, L' \in \mathcal{L}$, we say that C crosses (L, L') if there exist (not necessarily distinct) $u, v \in f^{-1}(L)$ and distinct $w, r \in f^{-1}(L')$ such that $\{u, w\}, \{v, r\} \in E(C)$. Moreover, we say that C crosses $\{L, L'\}$ if it crosses at least one of the pairs (L, L') and (L', L) .

That is, when an induced cycle C crosses $\{L, L'\}$, then there are exactly two edges from C which are across L and L' (i.e., one endpoint is in L and other is in L'). If C crosses $\{L, L'\}$, then by Observation 9.5, we know that for any “crossing edge” of $E(C)$ with one endpoint in L , the other must be in L' and vice versa. If an induced cycle C crosses $\{L, L'\}$ for two distinct cells $L, L' \in [t] \times [t']$, then at least 3 vertices of C are in $f^{-1}(L) \cup f^{-1}(L')$. Now, because of Observation 9.5 if $V(C) \cap f^{-1}(L) \neq \emptyset$ and there does not exist a cell L' such that C crosses $\{L, L'\}$, then the cell L interacts with two other cells. This case is captured in the following definition.

Definition 9.8 Given three distinct cells $L_1, L_2, L_3 \in \mathcal{L}$, we say that C crosses (L_1, L_2, L_3) if there exist $u \in f^{-1}(L_1)$ (not necessarily distinct) $v, w \in f^{-1}(L_2)$ and $r \in f^{-1}(L_3)$ such that $\{u, v\}, \{w, r\} \in E(C)$. Moreover, we say that C crosses $\{L_1, L_2, L_3\}$ if it crosses at least one of the tuples in $\{(L_s, L_r, L_t) : \{s, r, t\} = \{1, 2, 3\}\}$.

If an induced cycle C crosses $\{L_1, L_2, L_3\}$ for three distinct cells $(L_1, L_2, L_3) \in \mathcal{L}$, then at least one vertex of C is in $f^{-1}(L_r)$ for any $r \in [3]$. Next, we use the definitions above to capture the set of cycles which we would like to detect (if a solution exists).

Definition 9.9 A set \mathcal{C} of pairwise vertex-disjoint induced cycles is *simple* if it satisfies the following conditions:

- For every two distinct cells $L, L' \in \mathcal{L}$, there exist at most two cycles in \mathcal{C} that cross $\{L, L'\}$.
- For every three distinct cells $L_1, L_2, L_3 \in \mathcal{L}$, there exist at most two cycles in \mathcal{C} that cross $\{L_1, L_2, L_3\}$.

Given a cycle C (in G), denote $\text{cross}(C) = \{\{u, v\} \in E(C) : f(u) \neq f(v)\}$, and given a set \mathcal{C} of cycles, denote $\text{cross}(\mathcal{C}) = \bigcup_{C \in \mathcal{C}} \text{cross}(C)$. If $\{u, v\} \in \text{cross}(\mathcal{C})$ for some induced cycle C then either C crosses $\{f(u), f(v)\}$ or there exists a cell

$L_3 \in \mathcal{L} \setminus \{f(u), f(v)\}$ such that C crosses $\{f(u), f(v), L_3\}$. Next, we show that we can focus on deciding whether a simple set, rather than a general set, of k pairwise-disjoint cycles exists.

Lemma 9.10 *If (G, f, k) is a YES-instance, then G contains a simple set of k pairwise-disjoint induced cycles.*

Proof Suppose that $(G, f: V(G) \rightarrow \mathcal{L}, k)$ is a YES-instance. Next, let \mathcal{C} denote a set of k pairwise vertex-disjoint induced cycles that minimizes $|\text{cross}(\mathcal{C})|$ among all such sets of cycles (the existence of at least one such set of induced cycles is guaranteed by Observation 9.6). We will show that \mathcal{C} is simple. In what follows, we implicitly rely on Condition 1 in Definition 3.1.

First, suppose that there exist two distinct cells $L, L' \in \mathcal{L}$ and three cycles in \mathcal{C} that cross $\{L, L'\}$. Let C_1, C_2 and C_3 denote these three cycles. Notice that (a) for each $L^* \in \{L, L'\}$ and $i \in \{1, 2, 3\}$, $|V(C_i) \cap f^{-1}(L^*)| \geq 1$ and (b) $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap (f^{-1}(L) \cup f^{-1}(L'))| \geq 9$ (see Definition 9.7). Then, at least one of the three following conditions is true.

1. $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L)| = 3$. In this case $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L')| \geq 6$ (see statement (b) above). Here we replace C_1, C_2 and C_3 in \mathcal{C} by some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L)]$ and two vertex-disjoint cycles, each on three vertices, in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L')]$. We thus obtain a set of k pairwise vertex-disjoint induced cycles, \mathcal{C}' , such that $|\text{cross}(\mathcal{C}')| < |\text{cross}(\mathcal{C})|$, which is a contradiction to the choice of \mathcal{C} .
2. $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L')| = 3$: This case is symmetric to the previous one.
3. The above two cases do not occur. First we claim that there exist distinct $s, t \in \{1, 2, 3\}$ such that $|(V(C_s) \cup V(C_t)) \cap f^{-1}(L)| \geq 3$ and $|(V(C_s) \cup V(C_t)) \cap f^{-1}(L')| \geq 3$. Consider the cycles C_1 and C_2 . If C_1 and C_2 are the two required cycles, then $|V(C_i) \cap f^{-1}(L_1)| \geq 2$ and $|V(C_i) \cap f^{-1}(L_2)| = 1$ for $i \in \{1, 2\}$, where $\{L_1, L_2\} = \{L, L'\}$. If $|V(C_3) \cap f^{-1}(L_2)| \geq 2$, then C_1 and C_3 are the two required cycles. Otherwise we have that $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L_2)| = 3$. This is a contradiction to our assumption. In this case, we replace C_s and C_t in \mathcal{C} by some cycle on three vertices in $G[(V(C_s) \cup V(C_t)) \cap f^{-1}(L)]$ and some cycle on three vertices in $G[(V(C_s) \cup V(C_t)) \cap f^{-1}(L')]$. We thus obtain a set of k pairwise vertex-disjoint induced cycles, \mathcal{C}' , such that $|\text{cross}(\mathcal{C}')| < |\text{cross}(\mathcal{C})|$, which is a contradiction to the choice of \mathcal{C} .

Second, suppose that there exist three distinct cells $L_1, L_2, L_3 \in \mathcal{L}$ and three cycles in \mathcal{C} that cross $\{L_1, L_2, L_3\}$. Let C_1, C_2 and C_3 denote these three cycles. Then, it holds that $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L_1)| \geq 3$, $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L_2)| \geq 3$ and $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L_3)| \geq 3$. We replace C_1, C_2 and C_3 in \mathcal{C} by some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L_1)]$, some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L_2)]$, and some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(L_3)]$. This yields a contradiction to the choice of \mathcal{C} . □

Now, we examine the information given by Lemma 9.10 to extract a form that will be easier for us to exploit. To this end, we need the following. Given a set \mathcal{C} of

cycles and a cell $L \in \mathcal{L}$, denote $\text{cross}(\mathcal{C}, L) = (V(\text{cross}(\mathcal{C}))) \cap f^{-1}(L)$ (recall that $\text{cross}(\mathcal{C}) = \bigcup_{C \in \mathcal{C}} \text{cross}(C) = \{\{u, v\} \in E(C) : f(u) \neq f(v) \text{ and } C \in \mathcal{C}\}$).

Lemma 9.11 *If $(G, f: V(G) \rightarrow \mathcal{L}, k)$ is a YES-instance, then G contains a set \mathcal{C} of k pairwise-disjoint induced cycles such that for every cell $L \in \mathcal{L}$, it holds that $|\text{cross}(\mathcal{C}, L)| \leq 2304 = \mathcal{O}(1)$.*

Proof Suppose that (G, f, k) is a YES-instance. By Lemma 9.10, there exists a simple set \mathcal{C} of k pairwise-disjoint induced cycles. Let $L \in \mathcal{L}$ be some cell. Given a cell $L' \in \mathcal{L} \setminus \{L\}$, denote $\mathcal{C}(L') = \{C \in \mathcal{C} : C \text{ crosses } \{L, L'\}\}$. Moreover, given cells $L_2, L_3 \in \mathcal{L} \setminus \{L\}$, denote $\mathcal{C}(L_2, L_3) = \{C \in \mathcal{C} : C \text{ crosses } \{L, L_2, L_3\}\}$. Now, we define two sets of indices:

- $\mathcal{I} = \{L' \in \mathcal{L} \setminus \{L\} : \mathcal{C}(L') \neq \emptyset\}$.
- $\mathcal{I}' = \{(L_2, L_3) \in (\mathcal{L} \setminus \{L\}) \times (\mathcal{L} \setminus \{L\}) : \mathcal{C}(L_2, L_3) \neq \emptyset \text{ and } L_2 \neq L_3\}$.

Then, by Observation 9.5 and Lemma 9.10, it holds that

$$|\text{cross}(\mathcal{C}, L)| \leq 2 \left(\left| \bigcup_{L' \in \mathcal{I}} \mathcal{C}(L') \right| + \left| \bigcup_{(L_2, L_3) \in \mathcal{I}'} \mathcal{C}(L_2, L_3) \right| \right) \leq 4(|\mathcal{I}| + |\mathcal{I}'|).$$

Here, the first inequality follows from the fact that for any two cells L, L' and any induced cycle C , the number of edges of C with one endpoint in L and other in L' is at most 2 (by Observation 9.5). Note that for a cell $L = (i, j) \in \mathcal{L}$, $|\{(i', j') = L' \in \mathcal{L} \setminus \{(i, j)\} : |i - i'| \leq 2, |j - j'| \leq 2\}| \leq 24$. Thus, by Condition 2 in Definition 3.1, we have that $|\mathcal{I}| \leq 24$ and $|\mathcal{I}'| \leq 24 \cdot 23 = 552$. Therefore, $|\text{cross}(\mathcal{C}, L)| \leq 4(24 + 552) = 2304$. □

We are now ready to prove Lemma 9.3. Except for the arguments where we crucially rely on Lemma 9.11 and the fact that we have an $\mathcal{O}(\sqrt{k})$ -NCTD and not some general nice tree decomposition, the description of the DP is standard (see, e.g., [13]). Thus, we only give a sketch of the proof.

Proof (Sketch) of Lemma 9.3 Let us first examine a standard DP table \mathcal{A} to solve CYCLE PACKING when the parameter is $tw(G)$. Here, we have an entry $\mathcal{A}[v, \mathcal{Z}, k']$ for every node $v \in V(T)$, multiset \mathcal{Z} of subsets of sizes 1 or 2 of $\beta(v)$ and nonnegative integer $k' \leq k$. Moreover, each set of size 1 in \mathcal{Z} has only one occurrence and its vertex does not appear in any set of size 2 in \mathcal{Z} , and every vertex in $\beta(v)$ appears in at most two sets in \mathcal{Z} . Each such entry stores either 0 or 1. The value is 1 if and only if there exist a set \mathcal{S} of k' pairwise vertex-disjoint cycles in $G[\gamma(v)]$ and exists a set \mathcal{P} of internally pairwise vertex-disjoint paths in $G[\gamma(v)] - E(\beta(v))$ such that the following conditions are satisfied:

- $(\bigcup_{C \in \mathcal{S}} V(C)) \cap (\bigcup_{P \in \mathcal{P}} V(P)) = \emptyset$.
- On the one hand, for every cycle $C \in \mathcal{S}$, it holds that $|V(C) \cap \beta(v)| \leq 1$ and if $|V(C) \cap \beta(v)| = 1$ then there exists a set in \mathcal{Z} that is equal to $V(C) \cap \beta(v)$. On the other hand, if \mathcal{Z} contains a set of size 1, then there exists a cycle $C \in \mathcal{S}$ such that $V(C) \cap \beta(v)$ equals this set.

- On the one hand, for every path $P \in \mathcal{P}$, it holds that P contains at least three vertices, both endpoints of P belong to a distinct occurrence of a set in \mathcal{Z} (of size 2), and none of the internal vertices of P belongs to $\beta(v)$. On the other hand, for every occurrence X of a set of size 2 in \mathcal{Z} , there exists a distinct path P in \mathcal{P} such that the set containing the two endpoints of P is equal to X .

The intuitive meaning for \mathcal{Z} is as follows. We expect a set \mathcal{C} of k vertex-disjoint cycles in G with the following property. Let \mathcal{P}' be the restriction of \mathcal{C} on $G[\gamma(v)] - E(\beta(v))$. Let \mathcal{P} be the graph obtained after deleting isolated vertices from \mathcal{P}' . Then \mathcal{P} can be broken into (i) internally vertex-disjoint paths with endpoints in $\beta(v)$ and internal vertices in $\gamma(v) \setminus \beta(v)$, and (ii) k' vertex-disjoint cycles containing at most one vertex from $\beta(v)$ for each cycle. Then \mathcal{Z} will contain one set for each such path/cycle (intersecting $\beta(v)$) with the elements in the set being the vertices in the intersection of the path/cycle with $\beta(v)$.

The entry $\mathcal{A}[v, \mathcal{Z}, k']$ can be computed by examining all the entries $\mathcal{A}[u, \widehat{\mathcal{Z}}, \widehat{k}]$ where u is a child of v in T (recall that v can have at most two children). At the end of the computation of \mathcal{A} , we conclude that the input instance is a YES-instance if and only if $\mathcal{A}[r, \emptyset, k]$ contains 1 where r is the root of T . By Observation 9.4, we deduce that \mathcal{A} contains $2^{\mathcal{O}(k \log k)} \cdot n$ entries, where each entry can be computed in time $2^{\mathcal{O}(k \log k)}$.

Now we explain that in our case it is enough to compute subexponentially many entries unlike the standard DP. That is, we claim that for every $v \in V(T)$, it is sufficient to compute only $2^{\mathcal{O}(\sqrt{k} \log k)}$ entries. More precisely, for every $v \in V(T)$, it is sufficient to compute only entries $\mathcal{A}[v, \mathcal{Z}, k']$ such that $|\bigcup \mathcal{Z}| = \mathcal{O}(\sqrt{k})$ (there are only $2^{\mathcal{O}(\sqrt{k} \log k)}$ such entries). Indeed, suppose that the input instance is a YES-instance. Then, by Lemma 9.11, there exists a set \mathcal{C} of k pairwise vertex-disjoint induced cycles such that for every cell $L \in \mathcal{L}$, it holds that $|\text{cross}(\mathcal{C}, L)| = \mathcal{O}(1)$. Now, we sketch the main arguments that show that for every $v \in V(T)$, we still have an entry that “captures” \mathcal{C} (as explained below) and we are able to compute it in time $2^{\mathcal{O}(\sqrt{k} \log k)}$, which would imply that eventually, we would still be able to deduce that $\mathcal{A}[r, \emptyset, k]$ contains 1. For this purpose, consider some $v \in V(T)$. First, we notice that since for every cell $L \in \mathcal{L}$, it holds that $|\text{cross}(\mathcal{C}, L)| = \mathcal{O}(1)$, by Observation 9.5, and since T is an $\mathcal{O}(\sqrt{k})$ -NCTD, we have that there exists a set U of at most $\mathcal{O}(\sqrt{k})$ vertices in $\beta(v)$ such that every cycle $C \in \mathcal{C}$ satisfies at least one of the following conditions:

1. $V(C) \cap \beta(v) \subseteq U$,
2. $V(C) \subseteq \gamma(v) \setminus \beta(v)$,
3. $V(C) \subseteq V(G) \setminus \gamma(v)$,
4. $V(C) \subseteq f^{-1}(L) \subseteq \beta(v) \setminus U$, for some cell L .

Now, we let \mathcal{S} denote the set of cycles in \mathcal{C} such that all of their vertices, except at most one that belongs to $\beta(v)$, belong to $\gamma(v) \setminus \beta(v)$. Accordingly, we denote $k' = |\mathcal{S}|$. Moreover, let \mathcal{P} denote the set of every subpath of a cycle in \mathcal{C} (which is not fully contained in a cell) whose endpoints belong to $\beta(v)$ and whose set of internal vertices is a subset of size at least 1 of $\gamma(v) \setminus \beta(v)$. Finally, we define \mathcal{Z} as the multiset $\{\beta(v) \cap O \mid O \in \mathcal{S} \cup \mathcal{P}\}$. Then, it holds that $|\bigcup \mathcal{Z}| = \mathcal{O}(\sqrt{k})$ and \mathcal{C} witnesses that $\mathcal{A}[v, \mathcal{Z}, k']$ should be 1. Overall, by the existence of the set U that is mentioned

above, we conclude that the entry $\mathcal{A}[v, \mathcal{Z}, k']$ can be computed in time $2^{\mathcal{O}(\sqrt{k} \log k)}$. This completes the proof sketch. \square

10 Conclusion

In this paper, we gave subexponential algorithms of running time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ for a number of parameterized problems about cycles in unit disk graphs. The first natural question is whether the $\log k$ factor in the exponent can be shaved off. While we were not able to do it, we do not exclude such a possibility. In particular, it would be very interesting to build a theory for unit disk graphs, which is similar to the bidimensionality theory for planar graphs. In this context, it will be useful to provide a general characterization of parameterized problems admitting subexponential algorithms on unit disk graphs.

References


1. Alber, J., Fiala, J.: Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. In: Baeza-Yates, R., Montanari, U., Santoro, N. (eds.) Foundations of Information Technology in the Era of Network and Mobile Computing, pp. 26–37. Springer, New York (2002)
2. Alon, N., Yuster, R., Zwick, U.: Color-coding. *J. Assoc. Comput. Mach.* **42**(4), 844–856 (1995)
3. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.* **41**(1), 153–180 (1994)
4. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Narrow sieves for parameterized paths and packings. *J. Comput. Syst. Sci.* **87**, 119–139 (2017)
5. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25**(6), 1305–1317 (1996)
6. Bodlaender, H.L., Cygan, M., Kratsch, S., Nederlof, J.: Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.* **243**, 86–111 (2015)
7. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. *J. Comput. Syst. Sci.* **75**(8), 423–434 (2009)
8. Bodlaender, H.L., Drange, P.G., Dregi, M.S., Fomin, F.V., Lokshtanov, D., Pilipczuk, M.: A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.* **45**(2), 317–378 (2016)
9. Chan, T.M.: Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms* **46**(2), 178–189 (2003)
10. Chen, J., Kanj, I.A., Jia, W.: Vertex cover: further observations and further improvements. *J. Algorithms* **41**(2), 280–301 (2001)
11. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. *Discrete Math.* **86**(1–3), 165–177 (1990)
12. Clarkson, K.L., Varadarajan, K.R.: Improved approximation algorithms for geometric set cover. *Discrete Comput. Geom.* **37**(1), 43–58 (2007)
13. Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Cham (2015)
14. Demaine, E.D., Fomin, F.V., Hajiaghayi, M., Thilikos, D.M.: Subexponential parameterized algorithms on graphs of bounded genus and H -minor-free graphs. *J. ACM* **52**(6), 866–893 (2005)
15. Demaine, E.D., Hajiaghayi, M.: Bidimensionality: new connections between FPT algorithms and PTASs. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), pp. 590–601. ACM-SIAM, New York (2005)
16. Demaine, E.D., Hajiaghayi, M.: The bidimensionality theory and its algorithmic applications. *Comput. J.* **51**(3), 292–302 (2008)
17. Diestel, R.: Graph Theory. Graduate Texts in Mathematics, vol. 173, 4th edn. Springer, New York (2012)

18. Dorn, F., Penninkx, E., Bodlaender, H.L., Fomin, F.V.: Efficient exact algorithms on planar graphs: exploiting sphere cut decompositions. *Algorithmica* **58**(3), 790–810 (2010)
19. Dumitrescu, A., Pach, J.: Minimum clique partition in unit disk graphs. *Graphs Combin.* **27**(3), 399–411 (2011)
20. Fomin, F.V., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In: *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS 2016)*, pp. 515–524. IEEE Computer Society, Los Alamitos (2016)
21. Fomin, F.V., Lokshtanov, D., Panolan, F., Saurabh, S.: Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM* **63**(4), 29 (2016)
22. Fomin, F.V., Lokshtanov, D., Raman, V., Saurabh, S.: Bidimensionality and EPTAS. In: *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*, pp. 748–759. SIAM, Philadelphia (2011)
23. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Bidimensionality and geometric graphs. In: *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pp. 1563–1575. SIAM, Philadelphia (2012)
24. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pp. 503–510. SIAM, Philadelphia (2010)
25. Hale, W.K.: Frequency assignment: theory and applications. *Proc. IEEE* **68**(12), 1497–1514 (1980)
26. Har-Peled, S., Lee, M.: Weighted geometric set cover problems revisited. *J. Comput. Geom.* **3**(1), 65–85 (2012)
27. Har-Peled, S., Quanrud, K.: Approximation algorithms for polynomial-expansion and low-density graphs. In: *Proceedings of the 23rd Annual European Symposium (ESA 2015)*. Lecture Notes in Computer Science, vol. 9294, pp. 717–728. Springer, Heidelberg (2015)
28. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM* **32**(1), 130–136 (1985)
29. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM* **16**(6), 372–378 (1973)
30. Hunt III, H.B., Marathe, M.V., Radhakrishnan, V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms* **26**(2), 238–274 (1998)
31. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity. *J. Comput. Syst. Sci.* **63**(4), 512–530 (2001)
32. Ito, H., Kadoshita, M.: Tractability and intractability of problems on unit disk graphs parameterized by domain area. In: *Proceedings of the 9th International Symposium on Operations Research and Its Applications (ISORA'10)*, pp. 120–127 (2010)
33. Jansen, B.: Polynomial kernels for hard problems on disk graphs. In: *Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2010)*. Lecture Notes in Computer Science, vol. 6139, pp. 310–321. Springer, Berlin (2010)
34. Kammerlander, K.: C 900—an advanced mobile radio telephone system with optimum frequency utilization. *IEEE J. Sel. Areas Commun.* **2**(4), 589–597 (1984)
35. Koutis, I.: Faster algebraic algorithms for path and packing problems. In: *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*. Lecture Notes in Computer Science, vol. 5125, pp. 575–586. Springer, Berlin (2008)
36. Koutis, I., Williams, R.: Algebraic fingerprints for faster algorithms. *Commun. ACM* **59**(1), 98–105 (2016)
37. Marx, D.: Efficient approximation schemes for geometric problems? In: *Proceedings of the 13th Annual European Symposium on Algorithms (ESA 2005)*. Lecture Notes in Computer Science, vol. 3669, pp. 448–459. Springer, Berlin (2005)
38. Mustafa, N.H., Raman, R., Ray, S.: Settling the APX-hardness status for geometric set cover. In: *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS 2014)*, pp. 541–550. IEEE Computer Society, Philadelphia (2014)
39. Smith, W.D., Wormald, N.C.: Geometric separator theorems & applications. In: *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS 1998)*, pp. 232–243. IEEE Computer Society, Philadelphia (1998)
40. Thomassé, S.: A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms* **6**(2), 32 (2010)

41. Wang, D., Kuo, Y.-S.: A study on two geometric location problems. *Inf. Process. Lett.* **28**(6), 281–286 (1988)
42. Williams, R.: Finding paths of length k in $O^*(2^k)$ time. *Inf. Process. Lett.* **109**(6), 315–318 (2009)
43. Yeh, Y.S., Wilson, J., Schwartz, S.: Outage probability in mobile telephony with directive antennas and macrodiversity. *IEEE J. Sel. Areas Commun.* **2**(4), 507–511 (1984)
44. Zehavi, M.: Mixing color coding-related techniques. In: *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA 2015)*. Lecture Notes in Computer Science, vol. 9294, pp. 1037–1049. Springer, Heidelberg (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Fedor V. Fomin¹ · Daniel Lokshtanov^{1,2} · Fahad Panolan¹  · Saket Saurabh^{1,3} · Meirav Zehavi⁴

Fedor V. Fomin
fomin@ii.uib.no

Daniel Lokshtanov
daniello@ucsb.edu

Fahad Panolan
fahad.panolan@ii.uib.no

Saket Saurabh
saket@imsc.res.in

Meirav Zehavi
meiravze@bgu.ac.il

- ¹ Department of Informatics, University of Bergen, Bergen, Norway
- ² Present Address: University of California, Santa Barbara, USA
- ³ The Institute of Mathematical Sciences, HBNI, Chennai, India
- ⁴ Ben-Gurion University of the Negev, Beer-Sheva, Israel