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

Fedor V. Fomin[1], Daniel Lokshtanov[1], Fahad Panolan[1], Saket Saurabh[1,2], and Meirav Zehavi[1]

[1]Department of Informatics, University of Bergen, Norway, {fomin,daniello,fahadn.panolan,meirav.zehavi}@ii.uib.no
[2]The Institute of Mathematical Sciences, HBNI, Chennai, India, saket@imsc.res.in

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.

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 [26, 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 [12, 20, 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, 10, 25, 33, 39].

*Supported by Pareto-Optimal Parameterized Algorithms, ERC Starting Grant 715744, Parameterized Approximation, ERC Starting Grant 306992, and Rigorous Theory of Preprocessing, ERC Advanced Investigator Grant 267959.

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. T hroughout 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 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 [35, 36, 42, 4], matroid based techniques [23] for Longest Path and Longest Cycle, and kernelization techniques for Feedback Vertex Set [40]. We refer to [14] 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, Paturi and Zane [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. [15] with efficient algorithms on graphs of bounded treewidth [19], Longest Path, Longest Cycle, Feedback Vertex Set and Cycle Packing are solvable in time $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$ on planar graphs. The parameterized subexponential "tractability" of such problems can be extended to graphs excluding some fixed graph as a minor [17]. 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 $2^{\mathcal{O}(\sqrt{k}\log^2 k)}n^{\mathcal{O}(1)}$ on planar and apex-minor-free graphs [22]. The bidimensionality theory was also used to design (efficient) polynomial time approximation scheme ((E)PTAS) [16, 24] and polynomial kernelization [21] 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 [30, 28, 27, 29, 13, 38, 25]. 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 $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$ 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 [11] or Connected Vertex Cover [33] combined with an appropriate separation theorem [1, 10, 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 [25]. 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 [12] 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. 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 [8]. 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 $\mathsf{poly}(k)$ (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 $C_1, \ldots, C_d$, 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.

2 Preliminaries

For a positive integer t, we use $[t]$ as a shorthand for $\{1, 2, \ldots, t\}$. Given a function $f : A \to B$ and a subset $A' \subseteq A$, let $f|_{A'}$ denote the restriction of the function f to the domain A'. For a function

$f : A \to B$ and $B' \subseteq B$, $f^{-1}(B')$ denote 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 [18] 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 $E(G) = E(G[V(G) \setminus \{u, v\}]) \cup \{\{x_{\{u,v\}}, w\} \mid w \in N(\{u, v\})\}$. A graph H is called a *minor* of G, if H can be obtained from G by a sequence of edge deletion, edge contraction and vertex deletion. 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 $\{u_2, u_3, \dots, u_{\ell-1}\}$. 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 $[u_1 u_2 \dots u_\ell]$. 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 compete 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. For ease of presentation, for any function $f : D \to [a] \times [b]$, $i \in [a]$ 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 $\mathsf{pw}(G)$. 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 $\mathsf{tw}(G)$.

Proposition 2.3 ([9])**.** *Given a graph G and an integer k, in time $2^{\mathcal{O}(k)} \cdot n$, we can either decide that $\mathsf{tw}(G) > k$ 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 ([6])**.** *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, we say that a graph G *represents* O if each vertex in $V(G)$ represents a distinct geometric object in O, and every geometric object in O is represented by a distinct vertex in $V(G)$. In this case, we abuse notation and write $V(G) = O$. The *intersection graph of* O is a graph G that represent O and satisfies $E(G) = \{\{u, v\} : u, v \in O, u \cap v \neq \emptyset\}$.

Let $P = \{p_1 = (x_1, y_1), p_2 = (x_2, y_2), \ldots, p_n = (x_n, y_n)\}$ 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 $D = \{d_1, d_2, \ldots, d_n\}$. 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 $S = \{s_1, s_2, \ldots, s_n\}$. Then, the *unit square graph of* S is the intersection graph of S. Alternatively, the unit square graph of S is the geometric graph of 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 graph)**.** A graph G is a *clique-grid graph* if there exists a function $f : V(G) \to [t] \times [t']$, 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

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

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. 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 $y_{\max} = \max\{y_i \mid p_i = (x_i, y_i) \in D\}$. Accordingly, denote $\widehat{t} = \frac{x_{\max} - x_{\min}}{\sqrt{2}}$ and $\widehat{t'} = \frac{y_{\max} - y_{\min}}{\sqrt{2}}$. If $\widehat{t} = \lceil \widehat{t} \rceil$, then denote $t = \widehat{t} + 1$, and otherwise denote $t = \widehat{t}$. Similarly, if $\widehat{t'} = \lceil \widehat{t'} \rceil$, then denote $t' = \widehat{t'} + 1$, and otherwise denote $t' = \widehat{t'}$. Now, define $f : V(G) \to [t] \times [t']$ as follows. For all $p_i = (x_i, y_i) \in V(G)$, define $a_i = \lfloor \frac{x_i - x_{\min}}{\sqrt{2}} + 1 \rfloor$, $b_i = \lfloor \frac{y_i - y_{\min}}{\sqrt{2}} + 1 \rfloor$ 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, $\lfloor \frac{x_i - x_{\min}}{\sqrt{2}} + 1 \rfloor = \lfloor \frac{x_j - x_{\min}}{\sqrt{2}} + 1 \rfloor$ and $\lfloor \frac{y_i - y_{\min}}{\sqrt{2}} + 1 \rfloor = \lfloor \frac{y_j - y_{\min}}{\sqrt{2}} + 1 \rfloor$. 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

- $|\lfloor \frac{x_i - x_{\min}}{\sqrt{2}} \rfloor - \lfloor \frac{x_j - x_{\min}}{\sqrt{2}} \rfloor| \leq 2$, and

- $|\lfloor\frac{y_i - y_{\min}}{\sqrt{2}}\rfloor - \lfloor\frac{y_j - y_{\min}}{\sqrt{2}}\rfloor| \leq 2$.

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 $\lfloor\frac{x_i - x_{\min}}{\sqrt{2}}\rfloor - \lfloor\frac{x_j - x_{\min}}{\sqrt{2}}\rfloor \leq 2$. 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 $X = \frac{x_j - x_{\min}}{\sqrt{2}}$. Then, $\lfloor\frac{x_i - x_{\min}}{\sqrt{2}}\rfloor - \lfloor\frac{x_j - x_{\min}}{\sqrt{2}}\rfloor \leq \lfloor X + \sqrt{2}\rfloor - \lfloor X\rfloor \leq 2$. □

Similarly, we 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. 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 $y_{\max} = \max\{y_i \mid p_i = (x_i, y_i) \in D\}$. Accordingly, denote $\widehat{t} = x_{\max} - x_{\min}$ and $\widehat{t'} = y_{\max} - y_{\min}$. If $\widehat{t} = \lceil\widehat{t}\rceil$, then denote $t = \widehat{t} + 1$, and otherwise denote $t = \widehat{t}$. Similarly, if $\widehat{t'} = \lceil\widehat{t'}\rceil$, then denote $t' = \widehat{t'} + 1$, and otherwise denote $t' = \widehat{t'}$. Now, define $f : V(G) \to [t] \times [t']$ as follows. For all $p_i = (x_i, y_i) \in V(G)$, define $a_i = \lfloor x_i - x_{\min} + 1\rfloor$, $b_i = \lfloor y_i - y_{\min} + 1\rfloor$ 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, $\lfloor x_i - x_{\min} + 1\rfloor = \lfloor x_j - x_{\min} + 1\rfloor$ and $\lfloor y_i - y_{\min} + 1\rfloor = \lfloor y_j - y_{\min} + 1\rfloor$. Thus, we have that $|x_i - x_j| < 1$ and $|y_i - y_j| < 1$, 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 $|b_i - b_j| \leq 2$. In fact, we will actually prove that $|a_i - a_j| \leq 1$ and $|b_i - b_j| \leq 1$. By substituting a_i, a_j, b_i and b_j, it is sufficient to show that

- $|\lfloor x_i - x_{\min}\rfloor - \lfloor x_j - x_{\min}\rfloor| \leq 1$, and
- $|\lfloor y_i - y_{\min}\rfloor - \lfloor y_j - y_{\min}\rfloor| \leq 1$.

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 $\lfloor x_i - x_{\min}\rfloor - \lfloor x_j - x_{\min}\rfloor \leq 1$. Since G is the unit disk graph of D, $\{p_i, p_j\} \in E(G)$ and $x_j \leq x_i$, it holds that $x_i - x_j \leq 1$. Denote $X = x_j - x_{\min}$. Then, $\lfloor x_i - x_{\min}\rfloor - \lfloor x_j - x_{\min}\rfloor \leq \lfloor X + 1\rfloor - \lfloor X\rfloor \leq 1$. □

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, $(i_1, j_1), \ldots, (i_\ell, j_\ell)$, 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 $\beta(v) \setminus f^{-1}(i,j) = \beta(u) \setminus f^{-1}(i,j)$.
  - **Join**: v has exactly two children, u and w, and $\beta(v) = \beta(u) = \beta(w)$.

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 Section 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 $f : V(G) \to [t] \times [t']$, an induced subgraph H of G is a *backbone* for (G, f) if for every two distinct cells $(i,j), (i',j') \in [t] \times [t']$ for which there exist $u \in f^{-1}(i,j)$ and $v \in f^{-1}(i',j')$ such that $\{u,v\} \in E(G)$, there also exist $u' \in f^{-1}(i,j)$ and $v' \in f^{-1}(i',j')$ 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 $f^{-1}(i,j) \cap V(H) \leq |\{(i',j') \in [t] \times [t'] \setminus \{(i,j)\} \mid |i-i'| \leq 2, |j-j'| \leq 2\}| \leq 24$. 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

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

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 [25] 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 ([25])**.** *Any unit disk/square graph with maximum degree* Δ *contains a* $\frac{\mathsf{tw}}{100\Delta^3} \times \frac{\mathsf{tw}}{100\Delta^3}$ *grid as a minor.*

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* $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$*, one can either correctly conclude that* G *contains a* $\frac{\ell}{100 \cdot 599^3} \times \frac{\ell}{100 \cdot 599^3}$ *grid as a minor, or obtain a minimal backbone* H *for* (G, f) *with a nice tree decomposition* $\mathcal{T}$ *of width at most* $5k$*.*

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 $\frac{\ell}{100 \cdot 599^3} \times \frac{\ell}{100 \cdot 599^3}$ 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. $\square$

We use Lemma 4.5 with $\ell = \mathcal{O}(\sqrt{k})$. 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 $f : V(G) \to [t] \times [t']$, the *cell graph* of G, denoted by $\mathsf{cell}(G)$, 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* $\mathsf{cell}(G)$ *is a minor of* H*.*

Since for any graph G and a minor H of G, it holds that $\mathsf{tw}(H) \leq \mathsf{tw}(G)$, 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* $\mathsf{tw}(\mathsf{cell}(G)) \leq \mathsf{tw}(H)$*.*

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* $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$*, one can either correctly conclude that* G *contains a* $\frac{\ell}{100 \cdot 599^3} \times \frac{\ell}{100 \cdot 599^3}$ *grid as a minor, or compute a nice tree decomposition of* $\mathsf{cell}(G)$ *of width at most* 5ℓ*.*

Note that a nice tree decomposition of $\mathsf{cell}(G)$ 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* $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$*, one can either correctly conclude that* G *contains a* $\frac{\ell}{100 \cdot 599^3} \times \frac{\ell}{100 \cdot 599^3}$ *grid as a minor, or compute a* 5ℓ*-NCTD of* G*.*

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. 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 $f : V(G) \to [t] \times [t']$, 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 $V(H') \cap f^{-1}(i', j') \neq \emptyset$.

Definition 5.3. Let $I = (G, f : V(G) \to [t] \times [t'], k)$ 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 $\ell \geq 2k$.

We proceed by proving two claims concerning solutions to Longest Cycle and SI on clique-grid graphs.

Lemma 5.4. *Let $I = (G, f : V(G) \to [t] \times [t'], H, k)$ 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. Denote $i_{\min} = \min\{i \in [t] : (\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(H') \neq \emptyset\}$, $i_{\max} = \max\{i \in [t] : (\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(H') \neq \emptyset\}$, $j_{\min} = \min\{j \in [t'] : (\bigcup_{i \in [t]} f^{-1}(i, j)) \cap V(H') \neq \emptyset\}$ and $j_{\max} = \max\{j \in [t'] : (\bigcup_{i \in [t]} f^{-1}(i, j)) \cap V(H') \neq \emptyset\}$. To prove that H' is not $2k$-stretched, we need to prove that $i_{\max} - i_{\min} < 2k$ and $j_{\max} - j_{\min} < 2k$. We only prove that $i_{\max} - i_{\min} < 2k$, as the proof that $j_{\max} - j_{\min} < 2k$ is symmetric.

Let $i_1 < i_2 < \cdots < i_\ell$ 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 $i_1 = i_{\min}$ and $i_\ell = i_{\max}$. 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.

Now, since for all $r \in [\ell - 1]$, we proved that $i_{r+1} - i_r \leq 2$, we have that that there exist at least $\frac{i_{\max} - i_{\min}}{2} + 1$ indices $i \in [t]$ such that $\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(H') \neq \emptyset$. However, $|V(H)| \leq k$. Therefore $\frac{i_{\max} - i_{\min}}{2} + 1 \leq k$, which implies that $i_{\max} - i_{\min} < 2k$. □

Lemma 5.5. *Let $I = (G, f : V(G) \to [t] \times [t'], k)$ 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 there exists a flow of size 2 between them (see, e.g., [5]). 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 5.4, 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. □

Next, we prove a statement similar to the one of Theorem 5.1, but which concerns clique-grid graphs. Our method is inspired by Baker's technique [3].

Lemma 5.6. *Let $I = (G, f : V(G) \to [t] \times [t'], H, k)$ be an instance of* SI *(*LONGEST CYCLE*) on clique-grid 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 (G, f, H, k) is a* YES*-instance if and only if at least one instance in $\mathcal{I}$ is a* YES*-instance, and for all $(\widehat{G}, \widehat{f} : V(\widehat{G}) \to [\widehat{t}] \times [\widehat{t'}], \widehat{H}, \widehat{k}) \in \mathcal{I}$, it holds that $\widehat{G}$ is either an induced subgraph of G or $K_{\widehat{k}}$, $\widehat{t}, \widehat{t'} \leq 2k = \mathcal{O}(\widehat{k})$, $|\widehat{f}^{-1}(i, j)| < \widehat{k}$ for any cell $(i, j) \in [t] \times [t']$, $\widehat{H} = H$ and $\widehat{k} = k$.*

Proof. First, 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, $(K_k, \widehat{f} : V(K_k) \to [1] \times [1], H, k)$. 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, $(K_k, \widehat{f} : V(K_k) \to [1] \times [1], H, k)$. From now on, also suppose that if the input instance I is of LONGEST CYCLE, then it is not stretched .

Now, our kernelization algorithm works as follows. For every $(p, q) \in [t] \times [t']$, it computes

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

Accordingly, it computes $f_{p,q} : V(G_{p,q}) \to [\min\{2k, t\}] \times [\min\{2k, t'\}]$ as follows. For every $v \in V(G_{p,q})$, compute $f_{p,q}(v) = (i - p + 1, j - q + 1)$ where $(i, j) = f(v)$. Note that for all $(i, j) \in [\min\{2k, t\}] \times [\min\{2k, t'\}]$, it holds that $f^{-1}_{p,q}(i, j) = f^{-1}(i + p - 1, j + q - 1)$. Thus, since f is a representation of G, it holds that $f_{p,q}$ is a representation of $G_{p,q}$. Finally, our kernelization algorithm outputs $\mathcal{I} = \{I_{p,q} = (G_{p,q}, f_{p,q}, H, k) : (p, q) \in [t] \times [t']\}$.

To conclude the proof, it remains to 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 isomorhpic to H. Denote $i_{\min} = \min\{i \in [t] : (\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(H') \neq \emptyset\}$, $i_{\max} = \max\{i \in [t] : (\bigcup_{j \in [t']} f^{-1}(i, j)) \cap V(H') \neq \emptyset\}$, $j_{\min} = \min\{j \in [t'] : (\bigcup_{i \in [t]} f^{-1}(i, j)) \cap V(H') \neq \emptyset\}$ and

$j_{\max} = \max\{j \in [t'] : (\bigcup_{i\in[t]} f^{-1}(i,j)) \cap V(H') \neq \emptyset\}$. By Lemma 5.4, it holds that both $i_{\max} - i_{\min} < 2k$ and $j_{\max} - j_{\min} < 2k$. Therefore, H' is a subgraph of $G_{i_{\min},j_{\min}}$, which 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. Denote $i_{\min} = \min\{i \in [t] : (\bigcup_{j\in[t']} f^{-1}(i,j)) \cap V(C) \neq \emptyset\}$, $i_{\max} = \max\{i \in [t] : (\bigcup_{j\in[t']} f^{-1}(i,j)) \cap V(C) \neq \emptyset\}$, $j_{\min} = \min\{j \in [t'] : (\bigcup_{i\in[t]} f^{-1}(i,j)) \cap V(C) \neq \emptyset\}$ and $j_{\max} = \max\{j \in [t'] : (\bigcup_{i\in[t]} f^{-1}(i,j)) \cap V(C) \neq \emptyset\}$. Since (G, f, H, k) is not stretched, it holds that both $i_{\max} - i_{\min} < 2k$ and $j_{\max} - j_{\min} < 2k$. Therefore, C is a subgraph of $G_{i_{\min},j_{\min}}$, which implies that $I_{p,q}$ is a YES-instance. □

Towards proving Theorem 5.1, we extract a claim that is reused in Sections 6 and 7.

Corollary 5.7. *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 (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 $K_{\widehat{k}}$, $\widehat{t}, \widehat{t'} = \mathcal{O}(\widehat{k})$, $|\widehat{f}^{-1}(i,j)| < \widehat{k}$ for any cell $(i,j) \in [\widehat{t}] \times [\widehat{t'}]$, $\widehat{H} = H$ and $\widehat{k} = k$.*

Proof. 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). Then, by Lemma 5.6 with (G, f, H, k), we obtain (in polynomial time) the desired set $\mathcal{I}$. □

We are now ready to prove Theorem 5.1.

Proof of Theorem 5.1. First, by Corollary 5.7, we obtain (in polynomial time) a set $\mathcal{I}$ of instances of SI (LONGEST CYCLE) on clique-grid 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 $(\widehat{G}, \widehat{f}, \widehat{H}, \widehat{k}) \in \mathcal{I}$, it holds that $\widehat{G}$ is either an induced subgraph of G or $K_{\widehat{k}}$, $\widehat{t}, \widehat{t'} = \mathcal{O}(\widehat{k}^2)$, $|\widehat{f}^{-1}(i,j)| \leq \widehat{k}$ for a cell $(i,j) \in [\widehat{t}] \times [\widehat{t'}]$, $\widehat{H} = H$ and $\widehat{k} = k$. Thus, 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 $|V(\widehat{G})| = \mathcal{O}(k^3)$,$|E(\widehat{G})| = \mathcal{O}(k^4)$.

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 $|\widehat{f}^{-1}(i,j)| \leq k$ 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 $|E(\widehat{G})| = \mathcal{O}(k^4)$. □

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.*

Towards proving Theorem 6.1, we design an algorithm which given a clique-grid graph G along with its representation $f : V(G) \to [t] \times [t']$ and an integer k as input, runs in time $2^{\mathcal{O}(\sqrt{k}\log k)}|V(G)|^{\mathcal{O}(1)}$ and decides whether G has a cycle of length k or not. In Section 5 (Lemma 5.6) we have seen that SI admits a polynomial sized Turing kernel on clique-grid graphs. Hence to give an algorithm of running time $2^{\mathcal{O}(\sqrt{k}\log k)}|V(G)|^{\mathcal{O}(1)}$, we can restrict to instances of size bounded by polynomial in k. More precisely, because of Lemma 5.6, we can assume that the input to our algorithm is $(G, f : V(G) \to [t] \times [t'], k)$ where G is a clique-grid graph with a representation f, $|f^{-1}(i,j)| < k$ for all $(i,j) \in [t] \times [t']$ 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 $[t] \times [t'] \subseteq [2k] \times [2k]$.

Given an instance $(G, f : V(G) \to [2k] \times [2k], k)$, the algorithm applies a method inspired by Baker's technique [3] and obtains a family, $\mathscr{F}$, of $2^{\mathcal{O}(\sqrt{k}\log k)}$ instances of Exact k-Cycle. The family $\mathscr{F}$ has 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 $(G, f : V(G) \to [2k] \times [2k], k)$ is a Yes-instance if and only if there exists an instance $(H, f^* : V(H) \to [2k] \times [2k], k) \in \mathscr{F}$ which is a Yes-instance.
3. More over, for any instance $(H, f^* : V(H) \to [2k] \times [2k], k) \in \mathscr{F}$, H has a nice $7\sqrt{k}$-clique path decomposition ($7\sqrt{k}$-NCPD).

We will call the family $\mathscr{F}$ satisfying the above properties as *good family*. Let $(H, f^* : V(H) \to [2k] \times [2k], k)$ be an instance of $\mathscr{F}$. Let $\mathcal{P} = (X_1, \ldots, X_q)$ 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 (i,j) and (i',j') of f, the number of edges with one end point in (i,j) and other in (i',j') is at most 4. 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 $V(C)$ with one end point 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 $7\sqrt{k}$-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. As explained before, we assume that the number of vertices in the input graph is bounded by $k^{\mathcal{O}(1)}$.

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

Proof. Let C be a k length cycle 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 $2k$ columns of the $2k \times 2k$ grid with k blocks of two consecutive columns and label them from the set of labels $[\sqrt{k}]$. That is, each pair of columns $2i-1$ and $2i$, where $i \in [k]$ is labelled with $i \mod \sqrt{k}$. In other words both column $2i-1$ and $2i$ are labelled with $i \mod \sqrt{k}$. Then by pigeon hole principle there is a label $\ell \in \{1, 2, \ldots, \sqrt{k}\}$ 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 $2\sqrt{k}$ columns. That is, let $G_1 = G[\bigcup_{j=1}^{2(\ell-1)} f^{-1}(*, j)]$, and $G_{i+1} = G[\bigcup_{j=i\cdot 2\ell+1}^{\min\{i\cdot 2\ell+2\sqrt{k}, 2k\}} f^{-1}(*, j)]$ for all $i \in \{1, \ldots, \lceil\sqrt{k}\rceil\}$. Notice that G_i is clique-grid graph with representation $f_i : V(G_i) \to [2k] \times [2\sqrt{k}]$ defined as, $f_i(u) = (r, j)$, when $f(u) = (r, (i-1)2\ell + j)$. By the property 2 of clique-grid graph, $G \setminus (S \cup Y) = G_1 \uplus \ldots \uplus G_{\lceil\sqrt{k}\rceil+1}$.

Claim 6.3. *$G \setminus S$ has a nice $7\sqrt{k}$-clique path decomposition ($7\sqrt{k}$-NCPD).*

Proof. First, for each $i \in \{1, \ldots, \lceil\sqrt{k}\rceil + 1\}$, we define a path decomposition of G_i such that each bag is a union of at most $6\sqrt{k}$ many cells of f_i. As $G \setminus (S \cup Y) = G_1 \uplus \ldots \uplus G_{\lceil\sqrt{k}\rceil-1}$, 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 $\mathcal{P}_i = (X_{i,1}, X_{i,2}, \ldots X_{i,2k-2})$ 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 $(X_{i,1}, X_{i,2}, \ldots X_{i,2k-2})$. Hence $\mathcal{P}_i$ is a path decomposition of G_i. Since $X_{i,j} = f_i^{-1}(j, *) \cup f_i^{-1}(j+1, *) \cup f_i^{-1}(j+2, *)$, number of columns in f_i is at most $2\sqrt{k}$ and each cell of f_i is a cell of f, each $X_{i,j}$ is a union of $6\sqrt{k}$ many cells of f. Since $G \setminus (S \cup Y) = G_1 \uplus \ldots \uplus G_{\lceil\sqrt{k}\rceil+1}$, the sequence $\mathcal{P}' = (X_{1,1}, \ldots X_{1,2k-2}, X_{2,1}, \ldots X_{2,2k-2}, \ldots, X_{\lceil\sqrt{k}\rceil-1,1}, \ldots X_{\lceil\sqrt{k}\rceil-1,2k-2})$ is a path decomposition of $G \setminus (S \cup Y)$. More over, the vertices of each bag is a union of vertices from at most $6\sqrt{k}$ cells of f. Also, since $|Y| \leq \sqrt{k}$, the sequence $\mathcal{P} = (X_{1,1} \cup Y, \ldots X_{1,k-2} \cup Y, \ldots X_{\lceil\sqrt{k}\rceil-1,k-2} \cup Y)$ obtained by adding Y to each bag of $\mathcal{P}'$ we get a path decomposition of $G \setminus S$. More over, the vertices of 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 $\mathscr{F}$ as follows. For each $\ell \in \{1, \ldots, \lceil\sqrt{k}\rceil\}$ 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 $\mathscr{F}$. The number of choices of S and Y is bounded by $2^{\mathcal{O}(\sqrt{k}\log k)}$ and thus the size of $\mathscr{F}$ is bounded by $2^{\mathcal{O}(\sqrt{k}\log k)}$.

We claim that $\mathscr{F}$ is indeed a good family. Suppose there is a cycle C of length k in G. Then, by pigeon hole principle there is $\ell \in \{1, \ldots, \lceil\sqrt{k}\rceil\}$ 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 $\mathscr{F}$, is a YES instance. This completes the proof of the lemma. $\square$

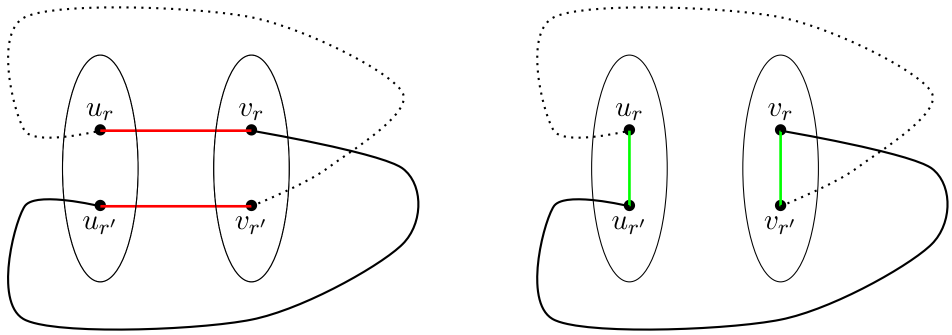

Figure 1: Illustration of Lemma 6.4. Figure on the left is the cycle $C = [u_r v_r]Q_1[u_{r'} v_{r'}]Q_2$ and the one on the right is the cycle $C' = [u_r u_{r'}]\overleftarrow{Q}_1[v_r v_{r'}]Q_2$

Now we can assume that we are solving Exact k-Cycle on (H, f, k), where $(H, f, k) \in \mathscr{F}$ (here we rename the function $f|_{V(H)}$ with f for ease of presentation). Now we 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 cells (i, j) and (i', j') of f, the number of edges of $E(C)$ with one end point in (i, j) and other (i', j') is at most 5.

Lemma 6.4. *Let $(H, f : V(H) \to [2k] \times [2k], k)$ be a* Yes *instance of* Exact k-Cycle*. Then there is a cycle C of length k in H such that for any two distinct cells (i, j) and (i', j') of f, the number of edges of $E(C)$ with one end point in (i, j) and other (i', j') is at most* 5.

Proof. Let C be a k length cycle such that the number edges of $E(C)$ whose end points are in different cells is minimized. We claim that for any two disjoint cells (i, j) and (i', j'), the number of edges of $E(C)$ with one end point in (i, j) and other (i', j') is at most 4. Suppose not. Then there exist (i, j) and (i', j') such that the number of edges of $E(C)$ with one end point in (i, j) and other in (i', j') is at least 6. Let $C = P_1[u_1v_1]P_2[u_2v_2]P_3[u_3v_3]P_4[u_4v_4]P_5[u_5v_5]P_6[u_6v_6]$ where for each $\{u_r, v_r\}, r \in [6]$, one end point is in the cell (i, j) and other in the cell $(i'j')$, and each subpath $P_\ell, \ell \in [6]$, can be empty too. Since C is a cycle, at least 3 edges from $\{\{u_r, v_r\} \ : \ i \in [6]\}$ 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 $\{r_1, r_2, r_3\} \subseteq [6]$. Then, by pigeon hole principle there exist $r, r' \in \{r_1, r_2, r_3\}$ such that either $u_r, u_{r'} \in f^{-1}(i, j)$ or $u_r, u_{r'} \in f^{-1}(i', j')$. Without loss of generality assume that $u_r, u_{r'} \in f^{-1}(i, j)$ (otherwise we rename cell (i, j) with (i', j') and vice versa). That is, $C = [u_r v_r]Q_1[u_{r'} v_{r'}]Q_2$ such that $u_r, u_{r'} \in f^{-1}(i, j)$ and $v_r, v_{r'} \in f^{-1}(i', j')$. Then, since $f^{-1}(i, j)$ and $f^{-1}(i', j')$ are cliques, $C' = [u_r u_{r'}]\overleftarrow{Q}_1[v_r v_{r'}]Q_2$ is a k length cycle in G, such that the number edges of $E(C')$ whose end points are in different cells is less than that of $E(C)$, which is contradiction to our assumption. See Fig. 1 for an illustration of C and C'. This completes the proof of the lemma. □

Next we design a DP algorithm that finds a cycle of length k, if it exists, satisfying properties of Lemma 6.4.

Lemma 6.5. *Let $(H, f : V(H) \to [2k] \times [2k], k) \in \mathscr{F}$ be an instance of* Exact k-Cycle *and $\mathcal{P}$ be a $7\sqrt{k}$-NCPD of H. Then, given $(H, f : V(H) \to [2k] \times [2k], k)$ 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 a cycle C in H such that for any two distinct cells (i, j) and (i', j') of f, the number of edges with one end point in (i, j) and other (i', j') is at most* 5*. Otherwise algorithm $\mathcal{A}$ will output* No.

Proof. Algorithm $\mathcal{A}$ is a DP algorithm over the $7\sqrt{k}$-NCPD $\mathcal{P} = (X_1, \ldots X_q)$ of H. For any $\ell \in [q]$, we define H_ℓ be the induced subgraph $H[\bigcup_{i \leq \ell} X_i]$ of H. Define $\mathcal{C}$ to be the set of k length cycles in H such that for any $C \in \mathcal{C}$ and two disjoint cells (i,j) and (i',j') of f, the number of edges of $E(C)$ with one end point in (i,j) and other (i',j') is at most 5. Let $C \in \mathcal{C}$. Since $\mathcal{P}$ is a $7\sqrt{k}$-NCPD and the fact that for any two distinct cells (i,j) and (i',j') of f, the number of edges of C with one end point in (i,j) and other (i',j') is at most 5, we have that for any bag X_ℓ of $\mathcal{P}$, the number of vertices of $V(C) \cap X_\ell$ which has a neighbour 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. Fix any $\ell \in [q]$ and define C_L the set of paths of C (or the cycle C itself) when we restrict C to H_ℓ. That is $C_L = H_\ell[E(C)]$. Let $\widehat{C}_L = \{\{u,v\} \mid \text{there is a } u\text{-}v \text{ path in } C_L\}$. Notice that $\bigcup_{P \in \widehat{C}_L} P$ is the set of vertices of degree 0 or 1 in C_L and $\bigcup_{P \in \widehat{C}_L} P \subseteq X_\ell$. Since X_ℓ is a union of vertices from at most $7\sqrt{k}$ many cells of f and for any two distinct cells (i,j) and (i',j') of f, the number of edges of $E(C)$ with one end point in (i,j) and other (i',j') is at most 5, and by property 2 of the clique-grid graph, we have that the cardinality of $\bigcup_{P \in \widehat{C}_L} P$ is at most $5 \cdot 24 \cdot 7\sqrt{k} = 840\sqrt{k}$. In our DP algorithm we will have state indexed by $(\ell, \widehat{C}_L, |E(C_L)|)$ 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 invariant.

> **Correctness Invariants:** (i) For every $C \in \mathcal{C}$, let $C_L = H_\ell[E(C)]$ and $\widehat{C}_L = \{\{u,v\} \mid \text{there is a connected component } P \text{ in } C_L \text{ and } P \text{ is a } u\text{-}v \text{ path as well}\}$. Then $\mathcal{A}[\ell, \widehat{C}_L, |E(C_L)|] = 1$, (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 $\mathcal{Q}$ of $|\mathcal{Z}|$ vertex disjoint paths in H_ℓ where the end points of each path are specified by a set in $\mathcal{Z}$ and $|E(\mathcal{Q})| = k'$, and (iii) If $\mathcal{A}[\ell, \emptyset, k] = 1$, then there is a cycle of length k in H_ℓ.

The correctness of the our algorithm will follow from the correctness invariant. Before explaining the computation of DP table entries, we first define some notations. Fix $\ell \in [q]$. For any $C \in \mathcal{C}$, define $C_L = H_\ell[E(C)]$ and $C_R = H[E(C) \setminus E(C_L)]$. For any family $\mathcal{Q}$ of vertex disjoint paths with end points in X_ℓ, define $\widehat{\mathcal{Q}} = \{\{u,v\} \mid \text{there is a } u\text{-}v \text{ path in } \mathcal{Q}\}$. That is, for any $C \in \mathcal{C}$, $\widehat{C}_L = \{\{u,v\} \mid \text{there is a } u\text{-}v \text{ path in } C_L\}$. Now we explain how to compute the values $\mathcal{A}[\cdot,\cdot,\cdot]$. 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 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 invariant holds for $\ell - 1$. When $\ell = 1$, $X_1 = \emptyset$ and the DP table entries are defined as follows.

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

Since $H_1 = (\emptyset, \emptyset)$, the correctness invarient follows. 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 an Introduce bag. That is $X_\ell = X_{\ell-1} \cup f^{-1}(i,j)$ for some cell (i,j). 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 set Q of vertex disjoint paths in $H[X_\ell \setminus X_{\ell-1}] = H[f^{-1}(i,j)]$ with at most 120 end points and $|E(\mathcal{Q})| = r$. Recall that that $\widehat{Q} = \{\{u,v\} \mid \text{there is a } u\text{-}v \text{ path in } Q\}$. Define $\widehat{\mathcal{Q}}_\ell(r) = \{\widehat{Q} \;:\; Q \in \mathcal{Q}_\ell(r)\}$.

Claim 6.6. *$|\widehat{\mathcal{Q}}_\ell(r)| = k^{\mathcal{O}(1)}$ and $\widehat{\mathcal{Q}}_\ell(r)$ can be enumerated in time $k^{\mathcal{O}(1)}$.*

Proof. We know that $H[X_\ell \setminus X_{\ell-1}]$ is a clique on $f^{-1}(i,j)$. For any family $\mathcal{W}$ of vertex disjoint sets of size at most 2 of $f^{-1}(i,j)$, one can get $|\mathcal{W}|$ vertex disjoint paths with end points being the one specified by the sets in $\mathcal{W}$ with total number of edges r, only if either $|\{W \in \mathcal{W} \ : \ |W| = 2\}| = r$ or $1 \leq |\{W \in \mathcal{W} \ : \ |W| = 2\}| < r$ and $|\{W \in \mathcal{W} \ : \ |W| = 2\}| + |f^{-1}(i,j) \setminus \bigcup_{W \in \mathcal{W}} W| \geq r$. Hence we can enumerate $\widehat{\mathcal{Q}}_\ell(r)$ in time $k^{\mathcal{O}(1)}$ □

For any family $\mathcal{Y}$ of sets of size at most 2, define $\mathcal{Y}' = \{A \in \mathcal{Y} \ : \ |A| = 2\}$. For any three families $\mathcal{Y}_1, \mathcal{Y}_2$ and $\mathcal{Y}_3$ of sets of size at most 2 in X_ℓ, we say that $\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \mathcal{Y}_3$ for a family of paths (respectively, a cycle) in $G' = K[X_\ell]$, if the graph $(\bigcup_{\substack{i \in [3] \\ Y \in \mathcal{Y}_i}} Y, \bigcup_{i \in [3]} \mathcal{Y}'_i)$ form a family of paths (respectively, a cycle). Consider the case when $\mathcal{Z} \neq \emptyset$.

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

Otherwise,

$$\begin{aligned} \mathcal{A}[\ell, \mathcal{Z}, k'] = \max \quad \Big\{ \quad & \mathcal{A}[\ell-1, \mathcal{Z}', k''] \ : \ \mathcal{Z}' \neq \emptyset \text{ and there exist } r \in \mathbb{N}, \widehat{\mathcal{Q}} \in \widehat{\mathcal{Q}}_\ell(r), \\ & E' \subseteq E\left(\bigcup_{Q \in \widehat{\mathcal{Q}}} Q, \bigcup_{Z \in \widehat{\mathcal{Z}'}} Z\right) \text{ such that } , |E'| \leq 120, k' = k'' + r + |E'|, \\ & \mathcal{Z}' \cup \widehat{\mathcal{Q}} \cup E' \text{ forms a set } \mathcal{R} \text{ of paths in } K[X_\ell], \text{ and } \widehat{\mathcal{R}} = \mathcal{Z} \Big\} \end{aligned} \tag{2}$$

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

$$\text{If } \mathcal{A}[\ell-1, \emptyset, k'] = 1, \text{ then we set } \mathcal{A}[\ell, \emptyset, k'] = 1 \tag{3}$$

Otherwise

$$\begin{aligned} \mathcal{A}[\ell, \emptyset, k'] = \max \quad \Big\{ \quad & \mathcal{A}[\ell-1, \mathcal{Z}', k''] \ : \ \text{there exist } r \in \mathbb{N}, \widehat{\mathcal{Q}} \in \widehat{\mathcal{Q}}_\ell(r), \\ & E' \subseteq E\left(\bigcup_{Q \in \widehat{\mathcal{Q}}} Q, \bigcup_{Z \in \widehat{\mathcal{Z}'}} Z\right) \text{ such that } , |E'| \leq 120, \\ & k' = k'' + r + |E'|, \text{ and } \mathcal{Z}' \cup \widehat{\mathcal{Q}} \cup E' \text{ forms a cycle in } K[X_\ell] \Big\} \end{aligned} \tag{4}$$

Notice that in the above computation (Equations 2 and 4) the number of potential choices for $\mathcal{Z}'$ is bounded by $2^{\mathcal{O}(\sqrt{k}\log k)}$. By Claim 6.6 we know that the cardinality of $\widehat{\mathcal{Q}}_\ell(r)$ 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' is 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 entries in time $2^{\mathcal{O}(\sqrt{k}\log k)}$.

Before proving the correctness invariant, we state the following simple claim, which can be proved using induction of ℓ.

Claim 6.7. *For any* $\ell \in [q]$, $\mathcal{A}[\ell, \emptyset, 0] = 1$.

Now we prove the correctness invariants. Let $C \in \mathcal{C}$. Recall that $C_L = H_\ell[E(C)]$ and $\widehat{C}_L = \{\{u,v\} \mid \text{there is a connected component } P \text{ in } C_L \text{ and } P \text{ is a } u\text{-}v \text{ path as well}\}$. Partition the edges C_L into $E_1 \uplus E_2 \uplus E'$ where $E_1 = E(C_L) \cap E(H_{\ell-1})$, $E_2 = E(C_L) \cap E(X_\ell \setminus X_{\ell-1})$

and $E' = E(C_L) \cap E(X_{\ell-1}, X_\ell \setminus X_{\ell-1})$. Let $C'_L = H[E_1]$. That is, $C'_L = H_{\ell-1}[E(C)]$. Let $\widehat{C'_L} = \{\{u,v\} \mid$ there is a u-v path in $C'_L\}$. We have two cases based whether $\widehat{C}_L = \emptyset$ or not.

Suppose $\widehat{C}_L \neq \emptyset$. If C_L is a subgraph of $H[X_\ell \setminus X_{\ell-1}]$, then $\widehat{C}_L \in \widehat{Q}(|E(C_L)|)$. Then by Equation 1, we have that $\mathcal{A}[\ell, \widehat{C}_L, |E(C_L)|] = 1$. 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 $E_2 \neq \emptyset$. In either case, we have that $\widehat{C'_L} \neq \emptyset$. Since $X_{\ell-1}$ is a union of vertices from $7\sqrt{k}$ cells, the property 2 of clique-grid graph and $C \in \mathcal{C}$, we have that the number of edges with one endpoint $X_{\ell-1}$ and other in $H \setminus X_{\ell-1}$ is at most $5 \cdot 24 \cdot 7\sqrt{k} = 840\sqrt{k}$. This implies that $|\bigcup_{D \in \widehat{C'_L}} D| \leq 840\sqrt{k}$. By the correctness invariant of statement (i) for $\ell - 1$, we have that (a) $\mathcal{A}[\ell-1, \widehat{C'_L}, |E(C'_L)|] = 1$. Consider the graph $H[E_2]$. The graph $H[E_2]$ is a collection $\mathcal{Q}$ of paths in $H[X_\ell \setminus X_{\ell-1}] = K[f^{-1}(i,j)]$. Since $C \in \mathcal{C}$, the number of edges of $E(C)$ with one end point in $f^{-1}(i,j)$ and other in $X_{\ell-1}$ is at most $4 \cdot 25 = 120$. This implies that (b) $|E'| \leq 120$ and (c) $\widehat{\mathcal{Q}} \in \widehat{Q}_\ell(r)$, where $r = |E(\mathcal{Q})|$. By facts (a), (b) and (c), using Equation 2, we get $\mathcal{A}[\ell, \widehat{C}_L, |E(C_L)|] = 1$.

Now consider the case when $\widehat{C}_L = \emptyset$. In this case either $E(C) \cap H_\ell = \emptyset$ or $E(C) \subseteq E(H_\ell)$. If $E(C) \cap H_\ell = \emptyset$, then $|E(C_L)| = 0$ and hence $\mathcal{A}[\ell, \widehat{C}_L, |E(C_L)|] = \mathcal{A}[\ell, \emptyset, 0] = 1$, by Claim 6.7. Now we have $\widehat{C}_L = \emptyset$ and $E(C) \subseteq E(H_\ell)$. This implies that $|E(C_L)| = k$. If $E(C_L) = E(C'_L)$, then $\mathcal{A}[\ell-1, \widehat{C'_L}, k] = \mathcal{A}[\ell-1, \emptyset, k] = 1$, by the statement (i) of the correctness invariant for $\ell-1$. So, now we have $E(C_L) \neq E(C'_L)$. This implies that either E_2 or E' is not empty. Since $|f^{-1}(i,j)| < k$ and $C_L = C$ form a cycle and $E(C_L) \neq E(C'_L)$, we have that $\widehat{C'_L} \neq \emptyset$. The graph $H[E_2]$ is a collection $\mathcal{Q}$ of paths in $H[X_\ell \setminus X_{\ell-1}] = K[f^{-1}(i,j)]$. As like before, we can bound (d) $|\bigcup_{D \in \widehat{C'_L}} D| \leq 840\sqrt{k}$, (e) $|E'| \leq 120$ and (g) $\widehat{\mathcal{Q}} \in \widehat{Q}_\ell(r)$, where $r = |E(\mathcal{Q})|$. By (d) and the statement (i) of the correctness invariant for $\ell-1$, we get (h) $\mathcal{A}[\ell-1, \widehat{C'_L}, |E(C'_L)|] = 1$. By facts (h), (e) and (d), using Equation 4, we get $\mathcal{A}[\ell, \widehat{C}_L, |E(C_L)|] = \mathcal{A}[\ell, \emptyset, k] = 1$. This completes the proof of statement (i).

Now we need to prove statement (ii) in 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 Equation 1 or because of Equation 2. If $\mathcal{A}[\ell, \mathcal{Z}, k']$ is set to 1 because of Equation 1, then we know that $\mathcal{Z} \in \widehat{\mathcal{Q}}_\ell(k')$. By the definition of $\widehat{\mathcal{Q}}_\ell(k')$, we get that there is a set $\mathcal{R}$ of vertex disjoint paths in $H[X_\ell \setminus X_{\ell-1}]$, hence in $H[X_\ell]$ and $\widehat{\mathcal{R}} = \mathcal{Z}$. So, now assume that $\mathcal{A}[\ell, \mathcal{Z}, k']$ is set to 1 because of Equation 2. This implies that there exist $k'', r \in \mathbb{N}$, a family $\mathcal{Z}'$ of vertex disjoint sets of size at most 2 of $X_{\ell-1}$, $\widehat{\mathcal{Q}} \in \widehat{\mathcal{Q}}_\ell(r)$, and $E' \subseteq E\left(\bigcup_{Q \in \widehat{\mathcal{Q}}} Q, \bigcup_{Z \in \widehat{\mathcal{Z}'}} Z\right)$ such that $\mathcal{A}[\ell-1, \mathcal{Z}', k''] = 1$, $|E'| \leq 120, k' = k'' + r + |E'|$, $\mathcal{Z}' \cup \widehat{\mathcal{Q}} \cup E'$ 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 the statement (ii) of the correctness invariant for $\ell-1$, we have that there is a set $\mathcal{Y}$ of $|\mathcal{Z}'|$ vertex disjoint paths in $H_{\ell-1}$ where the end points of each path are specified by a set in $\mathcal{Z}'$ and $|E(\mathcal{Y})| = k''$. Let $\mathcal{Q}$ be the set of paths in $\mathcal{Q}_\ell(r)$ corresponding to the set $\widehat{\mathcal{Q}}$. Thus by replacing each edge of $\mathcal{Z}'$ in $\mathcal{Z}' \cup \widehat{\mathcal{Q}} \cup E'$ by the corresponding path in $\mathcal{Y}$ and each edge of $\widehat{\mathcal{Q}}$ by a corresponding path in $\mathcal{Q}$, 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 interval vertices of paths in $\mathcal{Q}$ are disjoint from $X_{\ell-1} \cup \bigcup_{Z \in \mathcal{Z}} Z$. More over, $\widehat{\mathcal{W}} = \mathcal{Z}$ and $|E(\mathcal{W})| = |E(\mathcal{Y})| + |E(\mathcal{Q})| + |E'| = k'' + r + |E'| = k'$. This completes the proof of statement (ii) in the correctness invariant.

Now we prove statement (iii) in the correctness invariants. Suppose we set $A[\ell, \emptyset, k] = 1$. Then either $\mathcal{A}[\ell, \emptyset, k]$ is set to 1 because of Equation 3 or because of Equation 4. If $\mathcal{A}[\ell, \emptyset, k]$ is set to 1 because of Equation 1, then we know that $\mathcal{A}[\ell-1, \emptyset, k] = 1$, then by the statement (iii) of the correctness invariant 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 Equation 4. Then, there exist $k'', r \in \mathbb{N}$, a family $\mathcal{Z}'$ of vertex disjoint sets of size at most 2 of $X_{\ell-1}$, $\widehat{\mathcal{Q}} \in \widehat{\mathcal{Q}}_\ell(r)$, and $E' \subseteq E\left(\bigcup_{Q\in\widehat{\mathcal{Q}}} Q, \bigcup_{Z\in\widehat{\mathcal{Z}'}} Z\right)$ such that $\mathcal{A}[\ell-1, \mathcal{Z}', k''] = 1$, $|E'| \leq 120, k = k'' + r + |E'|$, $\mathcal{Z}' \cup \widehat{\mathcal{Q}} \cup E'$ forms a cycle in $K[X_\ell]$. Since $\mathcal{A}[\ell-1, \mathcal{Z}', k''] = 1$, by the statement (ii) of the correctness invariant for $\ell - 1$, we have that there is a set $\mathcal{Y}$ of $|\mathcal{Z}'|$ vertex disjoint paths in $H_{\ell-1}$ where the end points of each path are specified by a set in $\mathcal{Z}'$ and $|E(\mathcal{Y})| = k''$. Let $\mathcal{Q}$ be the set of paths in $\mathcal{Q}_\ell(r)$ corresponding to the set $\widehat{\mathcal{Q}}$. Thus by replacing each edge of $\mathcal{Z}'$ in $\mathcal{Z}' \cup \widehat{\mathcal{Q}} \cup E'$ by the corresponding path in $\mathcal{Y}$ and each edge of $\widehat{\mathcal{Q}}$ by a corresponding path in $\mathcal{Q}$, we can get a cycle C 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 interval vertices of paths in $\mathcal{Q}$ are disjoint from $X_{\ell-1} \cup \bigcup_{Z\in\mathcal{Z}} Z$. More over, $|E(C)| = |E(\mathcal{Y})| + |E(\mathcal{Q})| + |E'| = k'' + r + |E'| = k$. This completes the proof of statement (iii) in the correctness invariant.

Case 2: 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 $k' \in [k]$.

$$\mathcal{A}[\ell, \mathcal{Z}, k'] = \mathcal{A}[\ell-1, \mathcal{Z}, k'] \tag{5}$$

Clearly $\mathcal{A}[\ell, \mathcal{Z}, k']$ can be computed in $\mathcal{O}(1)$ time using the previously computed DP table entries.

Now we prove the correctness invariant. Let $C \in \mathcal{C}$. Recall that $C_L = H_\ell[E(C)]$ and $\widehat{C}_L = \{\{u, v\} \mid$ there is a connected component P in C_L and P is a u-v path as well$\}$. By arguments similar to those in Case 1, we get $|\bigcup_{D\in\widehat{C}_L} D| \leq 840\sqrt{k}$. Since $H_\ell = H_{\ell-1}$, we have that $C_L = H_{\ell-1}[E(C)]$. Hence by the correctness invariant for $\ell-1$, we have that $\mathcal{A}[\ell-1, \widehat{C}_L, |E(C_L)|] = 1$. Hence, by Equation 5, $\mathcal{A}[\ell-1, \widehat{C}_L, |E(C_L)|] = 1$.

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 $k' \in [k]$. Suppose in the above computation (Equation 5) we set $\mathcal{A}[\ell, \mathcal{Z}, k'] = 1$. This implies that $\mathcal{A}[\ell-1, \mathcal{Z}, k'] = 1$. Since $\mathcal{A}[\ell-1, \mathcal{Z}, k'] = 1$, by the correctness invariant for $\ell - 1$, we have that there is a set $\mathcal{Y}$ of $|\mathcal{Z}|$ vertex disjoint paths in $H_{\ell-1}$ where the end points of each path are specified by a set in $\mathcal{Z}$ and $|E(\mathcal{Y})| = k'$. Since $H_\ell = H_{\ell-1}$, $\mathcal{Y}$ is the required set of vertex disjoint paths and this completes the proof of statement (ii) in the correctness invariant.

Now we prove the statement (iii) in the correctness invariants. Suppose in the above computation (Equation 5) we set $\mathcal{A}[\ell, \emptyset, k] = 1$. This implies that $\mathcal{A}[\ell-1, \emptyset, k] = 1$ Since $\mathcal{A}[\ell-1, \emptyset, k] = 1$, by the correctness invariant for $\ell - 1$, we have that there is k length cycle in $H_{\ell-1}$, and hence in H_ℓ. This completes the proof of correctness invariants.

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

Lemmata 5.6, 6.2, 6.4 and 6.5 implies the following Lemma.

Lemma 6.8. Exact k-Cycle *on clique-grid graphs can be solved in time* $2^{\mathcal{O}(\sqrt{k}\log k)} n^{\mathcal{O}(1)}$.

Theorem 6.1 follows from Lemma 6.8 and Corollary 3.4. 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* $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$.

We start by stating a direct implication of Lemma 6.8.

Corollary 7.2. *Given a graph* G*, representation* f *and* $k \in \mathbb{N}$*, it can be determined in time* $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$ *whether* G *contain a cycle whose number of vertices is between* k *and* $2k$*.*

Proof. Run the algorithm given by Lemma 6.8 with $k = \ell, \ell+1, \ldots, 2\ell$, and return Yes if and only if at least one of the executions returns Yes. Correctness and running time follow directly from Lemma 6.8. □

Next, we examine the operation that contracts an edge. To this end, we need the following.

Definition 7.3. 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 $f_{/e} : V(G/e) \to [t] \times [t']$ as follows. For all $w \in V(G) \setminus \{v, u\}$, define $f_{/e}(w) = f(w)$. Moreover, define $f_{/e}(x_{\{u,v\}}) = f(u)$. By Definitions 3.1 and 4.6, we immediately have the following.

Observation 7.4. *The function* $f_{/e}$ *is a representation of* G/e*. Furthermore,* G *and* G/e *have the same cell graph.*

In particular, we deduce that $(G/e, f_{/e}, k)$ is an instance of Longest Cycle on clique-grid graphs. Next, we note that the operation that contracts an edge preserves the answer No—the correctness of this claim follows from the fact that G/e is a minor of G.

Observation 7.5. *Let* (G, f, k) *be an instance of* Longest Cycle *on clique-grid graphs. Then,* $(G/e, f_{/e}, k)$ *is an instance of* Longest Cycle *on clique-grid graphs such that if* (G, f, k) *is a* No*-instance, then* $(G/e, f_{/e}, k)$ *is a* No*-instance.*

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

Lemma 7.6. *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 $C = v_1 - v_2 - v_3 - \cdots - v_\ell - v_1$, 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 $v_{\ell-j}$, and obtain a cycle where this property holds). Now, note that $C' = x_{\{u,v\}} - v_2 - \cdots - v_{i-1} - x_{\{u,v\}}$ 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 isomorhpic to the cell graph of G. For the first case, we need the following result (see also [17, 14]).

Observation 7.7. *Let (G, k) be an instance of* Longest Cycle *on general graphs. If G contains a $\sqrt{k} \times \sqrt{k}$ grid as a minor, then (G, k) is a* Yes*-instance.*

For the second case, we need the following result (see also [14]).

Proposition 7.8 ([7])**.** Longest Cycle *on graphs of treewidth tw can be solved in time $2^{\mathcal{O}(tw)} \cdot n^{\mathcal{O}(1)}$.*

From Proposition 7.8 and the fact that $\mathsf{cell}(G)$ is a minor of G, we have the following.

Observation 7.9. *Let (G, f, k) be an instance of* Longest Cycle *on clique-grid graphs. Then, it can be determined in time $2^{\mathcal{O}(\mathsf{tw}(\mathsf{cell}(G)))} \cdot n^{\mathcal{O}(1)}$ whether $\mathsf{cell}(G)$ contains a cycle on at least k vertices, in which case (G, f, k) is a* Yes*-instance.*

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 3.4, we first obtain an equivalent instance (G, f, k) of Longest Cycle on clique-grid graphs (both instances refer to the same graph G and parameter k). Next, by using Lemma 4.9 with the parameter $\ell = 100 \cdot 599^3 \cdot \sqrt{k}$, we either correctly conclude that G contains a $\sqrt{k} \times \sqrt{k}$ grid as a minor, or compute a tree decomposition of $\mathsf{cell}(G)$ of width at most $500 \cdot 599^3 \cdot \sqrt{k} = \mathcal{O}(\sqrt{k})$. In the first case, by Observation 7.7, we are done. In the latter case, by using Observation 7.9, we determine in time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ whether $\mathsf{cell}(G)$ contains a cycle on at least k vertices, where if the answer is positive, then we are done. Thus, we next suppose that $\mathsf{cell}(G)$ 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 Corollary 7.2, we determine in time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$ whether G contain 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 $\{u, v\}$. By Lemma 7.6, we obtain an instance that is equivalent to the previous one. 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, on 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 answer No. □

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 observe that if we find a large grid, we can answer No (see also [17, 14]).

Observation 8.2. *Let (G, k) be an instance of* Feedback Vertex Set*. If G contains a $2\sqrt{k}\times 2\sqrt{k}$ grid as a minor, then (G, k) is a* No*-instance.*

This observation leads us to the following.

Lemma 8.3. *Let (G, O, k) be an instance of* Feedback Vertex Set *on unit disk/square graphs. Then, in time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot |V(G)|^{\mathcal{O}(1)}$, 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.*

Proof. First, by using Lemmata 3.2 or 3.3, we obtain a representation f of G. Then, by using Corollary 4.10 with $\ell = 200 \cdot 599^3 \cdot \sqrt{k} = \mathcal{O}(\sqrt{k})$, we either correctly conclude that G contains a $2\sqrt{k} \times 2\sqrt{k}$ grid as a minor, or compute an $\mathcal{O}(\sqrt{k})$-NCTD of G. In both cases, by Observation 8.2, we are done. □

Because of Lemma 8.3, to prove Theorem 8.1, we can focus on Feedback Vertex Set on clique-grid graphs, where the input also contains a $\mathcal{O}(\sqrt{k})$-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 and $\mathcal{T} = (T, \beta)$ is a $\mathcal{O}(\sqrt{k})$-NCTD of G. Notice that if there is a cell (i, j) of f, such that $|f^{-1}(i, j)| \geq k + 3$, then there is no feedback vertex set of size at most k in G, because $f^{-1}(i, j)$ is a clique of size at least $k + 3$ in G.

Observation 8.4. *Let $(G, f, k, \mathcal{T})$ be an instance of* Feedback Vertex Set*, where G is a clique-grid graph with representation f. If there is a cell (i, j) in f such that $|f^{-1}(i, j)| \geq k + 3$, then $(G, f, k, \mathcal{T})$ is a* No*-instance.*

The following observation follows from the fact that $\mathcal{T}$ is a $\mathcal{O}(\sqrt{k})$-NCTD and $|f^{-1}(i, j)| \leq k+2$ for any cell (i, j) of f.

Observation 8.5. *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 (i, j) in f, $|f^{-1}(i, j)| \leq k + 2$, 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 8.6. *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 Lemma 8.3 and Observations 8.4 and 8.6, 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 8.7. 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 $|f^{-1}(i,j)| \leq k+2$ for any cell (i,j) 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 $f^{-1}(i,j)$ for any cell (i,j). Now, the following claim follows from the fact that $\mathcal{T}$ is a $c\sqrt{k}$-NCTD and any solution contain at most 2 vertices from $f^{-1}(i,j)$ for any cell (i,j).

Claim 8.8. *For any $v \in V(T)$ and any solution S, $|S \cap \beta(v)| \leq 2c\sqrt{k}$.*

We first briefly explain what is the table entries in a standard DP algorithm for our problem on graphs of bounded treewidth [14]. 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 8.5 and Claim 8.8. That is, Observation 8.5 and Claim 8.8, shows that only $2^{\mathcal{O}(\sqrt{k}\log k)}|V(G)|^{\mathcal{O}(1)}$ many states in the DP table are relevant in our case. Recall that for any $v \in V(T)$, $\gamma(v)$ denote the union of the bags of v and its descendants. The standard DP table for our problem will have an entry indexed by $(v, U, U_1 \uplus U_2 \ldots U_\ell = U)$ where $v \in V(T)$, $U \subseteq \beta(v)$. The table entry $\mathcal{A}[v, U, U_1 \uplus U_2 \ldots U_\ell]$ 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 $\mathsf{tw}^{\mathcal{O}(\mathsf{tw})}|V(G)|^{\mathcal{O}(1)}$ 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 8.5 and Claim 8.8, 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 $(v, U, U_1 \uplus U_2 \ldots U_\ell = U)$ where $v \in V(T)$, $U \subseteq \beta(v)$ and $|U| \leq 2c\sqrt{k}$. Thus, the size of DP table, and hence the time to compute it takes $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$ 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 observe that if we find a large grid, we can answer YES (see also [17, 14]).

Observation 9.2. *Let (G, k) be an instance of* CYCLE PACKING *on general graphs. If G contains a $2\sqrt{k} \times 2\sqrt{k}$ grid as a minor, then (G, k) is a* YES*-instance.*

This observation leads us to the following.

Lemma 9.3. *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.*

Proof. First, by using Lemmata 3.2 or 3.3, we obtain a representation f of G. Then, by using Corollary 4.10 with $\ell = 200 \cdot 599^3 \cdot \sqrt{k} = \mathcal{O}(\sqrt{k})$, we either correctly conclude that G contains a $2\sqrt{k} \times 2\sqrt{k}$ grid as a minor, or compute an $\mathcal{O}(\sqrt{k})$-NCTD of G. In both cases, by Observation 9.2, we are done. ☐

Now, note that if there exists a cell $(i, j) \in [t] \times [t']$ such that $|f^{-1}(i, j)| \geq 3k$, then by Condition 1 in Definition 3.1, $G[f^{-1}(i, j)]$ is a clique on at least $3k$ vertices and thus it contains k pairwise vertex-disjoint cycles (triangles). More precisely, we have the following.

Observation 9.4. *Let $(G, f : V(G) \to [t] \times [t'], k)$ of* Cycle Packing *on clique-grid graphs. Then, if there exists a cell $(i, j) \in [t] \times [t']$ such that $|f^{-1}(i, j)| \geq 3k$, then (G, f, k) is a* Yes*-instance.*

By Lemma 9.3 and Observation 9.4, 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 9.5. 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 $(i, j) \in [t] \times [t']$, it holds that $|f^{-1}(i, j)| \leq 3k$.*

Let $(G, f : V(G) \to [t] \times [t'], k)$ denote the input instance of Cycle Packing, and let $\mathcal{T} = (T, \beta)$ denote our $\mathcal{O}(\sqrt{k})$-NCTD of G. Note that since $\mathcal{T}$ is an $\mathcal{O}(\sqrt{k})$-NCTD, and since for every $(i, j) \in [t] \times [t']$, it holds that $|f^{-1}(i, j)| \leq 3k$, we also have the following.

Observation 9.6. *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 9.7. *Let C be an induced cycle in G. Then, there does not exist a cell $(i, j) \in [t] \times [t']$ and two distinct vertices $u, v \in V(C) \cap f^{-1}(i, j)$ such that $\{u, v\} \notin E(C)$. In particular, for every cell $(i, j) \in [t] \times [t']$, exactly one of the following conditions holds.*

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

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

Definition 9.9. Given two distinct cells $(i, j), (i', j') \in [t] \times [t']$, we say that C *crosses* $((i, j), (i', j'))$ if there exist (not necessarily distinct) $u, v \in f^{-1}(i, j)$ and distinct $w, r \in f^{-1}(i', j')$ such that $\{u, w\}, \{v, r\} \in E(C)$. Moreover, we say that C *crosses* $\{(i, j), (i', j')\}$ if it crosses at least one of the pairs $((i, j), (i', j'))$ and $((i', j'), (i, j))$.

Definition 9.10. Given three distinct cells $(i_1, j_1), (i_2, j_2), (i_3, j_3) \in [t] \times [t']$, we say that C *crosses* $((i_1, j_1), (i_2, j_2), (i_3, j_3))$ if there exist $u \in f^{-1}(i_1, j_1)$, (not necessarily distinct) $v, w \in f^{-1}(i_2, j_2)$ and $r \in f^{-1}(i_3, j_3)$ such that $\{u, v\}, \{w, r\} \in E(C)$. Moreover, we say that C *crosses* $\{(i_1, j_1), (i_2, j_2), (i_3, j_3)\}$ if it crosses at least one of the tuples in $\{((i_s, j_s), (i_r, j_r), (i_t, j_t)) : \{s, r, t\} = \{1, 2, 3\}\}$.

Next, we use the definitions above to capture the set of cycles which we would like detect (if a solution exists).

Definition 9.11. A set $\mathcal{C}$ of pairwise vertex-disjoint induced cycles is *simple* if it satisfies the following conditions.

- For every two distinct cells $(i,j),(i',j') \in [t] \times [t']$, there exist at most two cycles in $\mathcal{C}$ that cross $\{(i,j),(i',j')\}$.
- For every three distinct cells $(i_1,j_1),(i_2,j_2),(i_3,j_3) \in [t] \times [t']$, there exist at most two cycles in $\mathcal{C}$ that cross $\{(i_1,j_1),(i_2,j_2),(i_3,j_3)\}$.

Given a cycle C (in G), denote $\mathrm{cross}(C) = \{\{u,v\} \in E(C) : f(u) \neq f(v)\}$, and given a set $\mathcal{C}$ of cycles, denote $\mathrm{cross}(\mathcal{C}) = \cup_{C\in\mathcal{C}}\mathrm{cross}(C)$. Next, we show that we can focus on the deciding whether a simple set, rather than a general set, of k pairwise-disjoint cycles exists.

Lemma 9.12. *If (G,f,k) is a* Yes*-instance, then G contains a simple set of k pairwise-disjoint induced cycles.*

Proof. Suppose that $(G, f : V(G) \to [t] \times [t'], k)$ is a Yes-instance. Next, let $\mathcal{C}$ denote a set of k pairwise vertex-disjoint induced cycles that minimizes $|\mathrm{cross}(\mathcal{C})|$ among all such sets of cycles (the existence of at least one such set of induced cycles is guaranteed by Observation 9.8). 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 $(i,j),(i',j') \in [t] \times [t']$ and three cycles in $\mathcal{C}$ that cross $\{(i,j),(i',j')\}$. Let C_1, C_2 and C_3 denote these three cycles. Then, at least one of the three following conditions is true.

1. There exist distinct $s,t \in \{1,2,3\}$ such that $|(V(C_s) \cup V(C_t)) \cap f^{-1}(i,j)| \geq 3$ and $|(V(C_s) \cup V(C_t)) \cap f^{-1}(i',j')| \geq 3$: In this case, we replace C_s and C_t in $\mathcal{C}$ by some cycle on three vertices in $G[(V(C_s) \cup V(C_t)) \cap f^{-1}(i,j)]$ and some cycle on three vertices in $G[(V(C_s) \cup V(C_t)) \cap f^{-1}(i',j')]$. We thus obtain a set of k pairwise vertex-disjoint cycles, $\mathcal{C}'$, such that $|\mathrm{cross}(\mathcal{C}')| < |\mathrm{cross}(\mathcal{C})|$, which is a contradiction to the choice of $\mathcal{C}$.
2. $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i,j)| = 3$ and $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i',j')| \geq 6$: In this case, we replace C_1, C_2 and C_3 in $\mathcal{C}$ by some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i,j)]$ and two vertex-disjoint cycles, each on three vertices, in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i',j')]$. We thus obtain a set of k pairwise vertex-disjoint cycles, $\mathcal{C}'$, such that $|\mathrm{cross}(\mathcal{C}')| < |\mathrm{cross}(\mathcal{C})|$, which is a contradiction to the choice of $\mathcal{C}$.
3. $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i,j)| \geq 6$ and $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i',j')| = 3$: This case is symmetric to the previous one.

Second, suppose that there exist three distinct cells $(i_1,j_1),(i_2,j_2),(i_3,j_3) \in [t] \times [t']$ and three cycles in $\mathcal{C}$ that cross $\{(i_1,j_1),(i_2,j_2),(i_3,j_3)\}$. Let C_1, C_2 and C_3 denote these three cycles. Then, it holds that $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i_1,j_1)| \geq 3$, $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i_2,j_2)| \geq 3$ and $|(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i_3,j_3)| \geq 3$. We replace C_1, C_2 and C_3 in $\mathcal{C}$ by some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i_1,j_1)]$, some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i_2,j_2)]$, and some cycle on three vertices in $G[(V(C_1) \cup V(C_2) \cup V(C_3)) \cap f^{-1}(i_3,j_3)]$. □

Now, we examine the information given by Lemma 9.12 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 $(i,j) \in [t] \times [t']$, denote $\mathrm{cross}(\mathcal{C},i,j) = (\bigcup \mathrm{cross}(\mathcal{C})) \cap f^{-1}(i,j)$ (note that $\bigcup \mathrm{cross}(\mathcal{C})$ is the set of every vertex that is an endpoint of an edge in $\mathrm{cross}(\mathcal{C})$).

Lemma 9.13. *If $(G, f : V(G) \to [t] \times [t'], k)$ is a* Yes*-instance, then G contains a set $\mathcal{C}$ of k pairwise-disjoint induced cycles such that for every cell $(i,j) \in [t] \times [t']$, it holds that $|\text{cross}(\mathcal{C}, i, j)| \leq 2304 = \mathcal{O}(1)$.*

Proof. Suppose that (G, f, k) is a Yes-instance. By Lemma 9.12, there exists a simple set $\mathcal{C}$ of k pairwise-disjoint induced cycles. Let $(i,j) \in [t] \times [t']$ be some cell. Given a cell $(i',j') \in [t] \times [t'] \setminus \{(i,j)\}$, denote $\mathcal{C}(i',j') = \{C \in \mathcal{C} : C \text{ crosses } \{(i,j),(i',j')\}\}$. Moreover, given cells $(i_2,j_2),(i_3,j_3) \in [t] \times [t'] \setminus \{(i,j)\}$, denote $\mathcal{C}(i_2,j_2,i_3,j_3) = \{C \in \mathcal{C} : C \text{ crosses } \{(i,j),(i_2,j_2),(i_3,j_3)\}\}$. Now, we define two sets of indices:

- $\mathcal{I} = \{(i',j') \in [t] \times [t'] \setminus \{(i,j)\} : \mathcal{C}(i',j') \neq \emptyset\}$.
- $\mathcal{I}' = \{((i_2,j_2),(i_3,j_3)) \in ([t] \times [t'] \setminus \{(i,j)\}) \times ([t] \times [t'] \setminus \{(i,j),(i_2,j_2)\}) : \mathcal{C}(i_2,j_2,i_3,j_3) \neq \emptyset\}$.

Then, by Observation 9.7 and Lemma 9.12, it holds that

$$\begin{aligned} |\text{cross}(\mathcal{C}, i, j)| &\leq 2(|\bigcup_{(i',j') \in \mathcal{I}} \mathcal{C}(i',j')| + |\bigcup_{((i_2,j_2),(i_3,j_3)) \in \mathcal{I}'} \mathcal{C}(i_2,j_2,i_3,j_3)|) \\ &\leq 4(|\mathcal{I}| + |\mathcal{I}'|). \end{aligned}$$

Note that $|\{(i',j') \in [t] \times [t'] \setminus \{(i,j)\} \mid |i-i'| \leq 2, |j-j'| \leq 2\}| \leq 24$. Thus, by Condition 2 in Definition 3.1, we have that $|\mathcal{I}| \leq 24$ and $|\mathcal{I}'| \leq 24 \cdot 23 = 552$. Therefore, $\text{cross}(\mathcal{C}, i, j)| \leq 4(24 + 552) = 2304$. ☐

We are now ready to prove Lemma 9.5. Except for the arguments where we crucially rely on Lemma 9.13 and the fact the 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., [14]). Thus, we only give a sketch of the proof.

Proof sketch of Lemma 9.5. Let us first examine a standard DP table $\mathcal{A}$ to solve Cycle Packing when the parameter is $\mathsf{tw}(G)$. 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 an only if there exist a set $\mathcal{S}$ of k' pairwise vertex-disjoint cycles in $G[\gamma(v)]$ and a set $\mathcal{P}$ of internally pairwise vertex-disjoint paths in $G[\gamma(v)]$ 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 $C \in \mathcal{C}$, 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 $C \in \mathcal{C}$ 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 entry $\mathcal{A}[v, \mathcal{Z}, k']$ can be computed by examining the all 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 9.6, we deduce that $\mathcal{A}$ contains $2^{\mathcal{O}(k \log k)} \cdot n$ entries, where each entry can be computed in time $2^{\mathcal{O}(k \log k)}$.

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 9.13, there exists a set $\mathcal{C}$ of k pairwise vertex-disjoint induced cycles such that for every cell $(i, j) \in [t] \times [t']$, it holds that $|\text{cross}(\mathcal{C}, i, j)| = \mathcal{O}(1)$. 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 $(i, j) \in [t] \times [t']$, it holds that $|\text{cross}(\mathcal{C}, i, j)| = \mathcal{O}(1)$, by Observation 9.7, and since $\mathcal{T}$ 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. $V(C) \subseteq V(G) \setminus \gamma(v)$.

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 $\mathcal{C}$ 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 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. ☐

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] Jochen Alber and Jiří Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. In *Foundations of Information Technology in the Era of Network and Mobile Computing*, pages 26–37. Springer, 2002.

[2] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. Assoc. Comput. Mach.*, 42(4):844–856, 1995.

[3] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.*, 41(1):153–180, 1994.

[4] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *CoRR*, abs/1007.1161, 2010.

[5] Andreas Björklund, Thore Husfeldt, and Nina Taslaman. Shortest cycle through specified elements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1747–1753, 2012.

[6] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[7] Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015.

[8] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[9] Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.

[10] Timothy M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, 2003.

[11] Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: further observations and further improvements. *Journal of Algorithms*, 41(2):280–301, 2001.

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

[13] Kenneth L. Clarkson and Kasturi R. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete & Computational Geometry*, 37(1):43–58, 2007.

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

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

[16] Erik D. Demaine and MohammadTaghi Hajiaghayi. Bidimensionality: new connections between fpt algorithms and ptass. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, pages 590–601, New York, 2005. ACM-SIAM.

[17] Erik D. Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *Comput. J.*, 51(3):292–302, 2008.

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

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

[20] Adrian Dumitrescu and János Pach. Minimum clique partition in unit disk graphs. *Graphs and Combinatorics*, 27(3):399–411, 2011.

[21] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pages 503–510. SIAM, 2010.

[22] Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 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), to appear*, 2016.

[23] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29, 2016.

[24] Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Bidimensionality and EPTAS. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 748–759, 2011.

[25] Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1563–1575. SIAM, 2012.

[26] William K Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1514, 1980.

[27] Sariel Har-Peled and Mira Lee. Weighted geometric set cover problems revisited. *JoCG*, 3(1):65–85, 2012.

[28] Sariel Har-Peled and Kent Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294, pages 717–728. Springer, 2015.

[29] Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.

[30] Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, S. S. Ravi, Daniel J. Rosenkrantz, and Richard Edwin Stearns. Nc-approximation schemes for NP- and pspace-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998.

[31] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity. *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[32] Hiro Ito and Masakazu Kadoshita. 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 (ISORA10)*, pages 120–127, 2010.

[33] Bart M. P. Jansen. Polynomial kernels for hard problems on disk graphs. In *Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 6139 of *Lecture Notes in Comput. Sci.*, pages 310–321. Springer, 2010.

[34] KARL Kammerlander. C 900-an advanced mobile radio telephone system with optimum frequency utilization. *IEEE journal on selected areas in communications*, 2(4):589–597, 1984.

[35] Ioannis Koutis. Faster algebraic algorithms for path and packing problems. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, volume 5125 of *Lecture Notes in Computer Science*, pages 575–586, 2008.

[36] Ioannis Koutis and Ryan Williams. Algebraic fingerprints for faster algorithms. *Commun. ACM*, 59(1):98–105, 2016.

[37] Dániel Marx. Efficient approximation schemes for geometric problems? In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, volume 3669 of *Lecture Notes in Comput. Sci.*, pages 448–459. Springer, 2005.

[38] Nabil H. Mustafa, Rajiv Raman, and Saurabh Ray. Settling the apx-hardness status for geometric set cover. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 541–550. IEEE Computer Society, 2014.

[39] Warren D. Smith and Nicholas C. Wormald. Geometric separator theorems & applications. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 232–243. IEEE Computer Society, 1998.

[40] Stéphan Thomassé. A quadratic kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.

[41] DW Wang and Yue-Sun Kuo. A study on two geometric location problems. *Information processing letters*, 28(6):281–286, 1988.

[42] Ryan Williams. Finding paths of length k in $O^*(2^k)$ time. *Inf. Process. Lett.*, 109(6):315–318, 2009.

[43] Yu-Shuan Yeh, J Wilson, and S Schwartz. Outage probability in mobile telephony with directive antennas and macrodiversity. *IEEE journal on selected areas in communications*, 2(4):507–511, 1984.

[44] Meirav Zehavi. Mixing color coding-related techniques. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA)*, volume 9294 of *Lecture Notes in Comput. Sci.*, pages 1037–1049. Springer, 2013.