Parameterized Complexity of Non-Separating and Non-Disconnecting Paths and Sets

Ankit Abhinav
National Institute of Science, Education and Research, An OCC of Homi Bhabha National Institute, Bhubaneswar, Odisha, India

Susobhan Bandopadhyay
National Institute of Science, Education and Research, An OCC of Homi Bhabha National Institute, Bhubaneswar, Odisha, India

Aritra Banik
National Institute of Science, Education and Research, An OCC of Homi Bhabha National Institute, Bhubaneswar, Odisha, India

Yasuaki Kobayashi
Kyoto University, Kyoto, Japan

Shunsuke Nagano
Kyoto University, Kyoto, Japan

Yota Otachi
Nagoya University, Nagoya, Japan

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India

Abstract

For a connected graph $G = (V, E)$ and $s, t \in V$, a non-separating s-t path is a path P between s and t such that the set of vertices of P does not separate G, that is, $G - V(P)$ is connected. An s-t path P is non-disconnecting if $G - E(P)$ is connected. The problems of finding shortest non-separating and non-disconnecting paths are both known to be NP-hard. In this paper, we consider the problems from the viewpoint of parameterized complexity. We show that the problem of finding a non-separating s-t path of length at most k is W[1]-hard parameterized by k, while the non-disconnecting counterpart is fixed-parameter tractable (FPT) parameterized by k. We also consider the shortest non-separating path problem on several classes of graphs and show that this problem is NP-hard even on bipartite graphs, split graphs, and planar graphs. As for positive results, the shortest non-separating path problem is FPT parameterized by k on planar graphs and on unit disk graphs (where no s, t is given). Further, we give a polynomial-time algorithm on chordal graphs if k is the distance of the shortest path between s and t.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Non-separating path, Parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.6

Funding *Yasuaki Kobayashi*: JSPS KAKENHI Grant Numbers JP20H05793, JP20K19742, JP21H03499
Yota Otachi: JSPS KAKENHI Grant Numbers JP18H04091, JP18K11168, JP18K11169, JP20H05793, JP21K11752.

1 Introduction

Lovász's path removal conjecture states the following claim: There is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every $f(k)$-connected graph G and every pair of vertices u and v, G has a path P between u and v such that $G - V(P)$ is k-connected. This claim still remains open, while

47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022).
Editors: Stefan Szeider, Robert Ganian, and Alexandra Silva; Article No. 6; pp. 6:1–6:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

some special cases have been resolved [4, 15, 17, 23]. Tutte [23] proved that $f(1) = 3$, that is, every triconnected graph satisfies that for every pair of vertices, there is a path between them whose removal results a connected graph. Kawarabayashi et al. [15] proved a weaker version of this conjecture: There is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that for every $f(k)$-connected graph G and every pair of vertices u and v, G has an induced path P between u and v such that $G - E(P)$ is k-connected.

As a practical application, let us consider a network represented by an undirected graph G, and we would like to build a private channel between a specific pair of nodes s and t. For some security reasons, the path used in this channel should be dedicated to the pair s and t, and hence all other connections do not use all nodes and/or edges on this path while keeping their connections. In graph-theoretic terms, the vertices (resp. edges) of the path between s and t does not form a separator (resp. a cut) of G. Tutte's result [23] indicates that such a path always exists in triconnected graphs, but may not exist in biconnected graphs. In addition to this connectivity constraint, the path between s and t is preferred to be short due to the cost of building a private channel. Motivated by such a natural application, the following two problems are studied.

▶ **Definition 1.** *Given a connected graph G, $s, t \in V(G)$, and an integer k,* Shortest Non-Separating Path *asks whether there is a path P between s and t in G such that the length of P is at most k and $G - V(P)$ is connected. When s and t are not part of the input, and we want to find a path P of length k, such that $G - V(P)$ is connected, then we call the problem* Terminal Independent Shortest Non-Separating Path (TI-Shortest Non-Separating Path)*.*

▶ **Definition 2.** *Given a connected graph G, $s, t \in V(G)$, and an integer k, Shortest Non-Disconnecting Path asks whether there is a path P between s and t in G such that the length of P is at most k and $G - E(P)$ is connected.*

Given, the Shortest Non-Separating Path problem, a natural question arises about the complexity of the problem, when in the problem we replace the demand of P being a path with P being a connected set. Given a connected graph G, and an integer k, Smallest Non-Separating Set asks whether there is a vertex subset X of size k in G such that $G[X]$ is connected and $G - X$ is connected. Similarly, an edge counterpart can be defined as Smallest Non-Disconnecting Set. An edge set X is said to be connected if the graph $G' = (V(X), X)$ is connected. We say that a path P is *non-separating* (in G) if $G - V(P)$ is connected and is *non-disconnecting* (in G) if $G - E(P)$ is connected. Similarly, we define the notion of *non-separating* set and *non-disconnecting* set.

1.1 Our Results and Methods

We investigate the parameterized complexity of above problems and obtain following results.

1. Shortest Non-Separating Path and Smallest Non-Separating Set are W[1]-hard. These are obtained by parameterized reductions from Multicolored Clique and Clique, respectively.
2. Shortest Non-Disconnecting Path and Smallest Non-Disconnecting Set are FPT parameterized by k. These algorithms are based on matroid based tools used in parameterized complexity [11]. In particular, given a graph G, there is a well-known matroid, defined by ground set being $E(G)$ and the family of independent sets being a subsets of Y of $E(G)$, such that $G - Y$ is connected. These are called cographic matroid.

A crucial observation for the FPT algorithms for Shortest Non-Disconnecting Path and Smallest Non-Disconnecting Set is that the set of edges in a non-disconnecting path or non-disconnecting set can be seen as an independent set of a cographic matroid. By applying the representative family of matroids [11], we show that Shortest Non-Disconnecting Path and Smallest Non-Disconnecting Set can be solved in $2^{\omega k}|V|^{O(1)}$ time, where ω is the exponent of the matrix multiplication. We also show that Shortest Non-Disconnecting Path is OR-compositional, that is, there is no polynomial kernel unless coNP $\subseteq$ NP/poly.

3. To cope with the intractability of Shortest Non-Separating Path, we consider the problem on planar graphs and unit disk graphs and show that Shortest Non-Separating Path is FPT parameterized by k on planar graphs and TI-Shortest Non-Separating Path is FPT parameterized by k on unit disk graphs. The result on planar graphs can be generalized to wider classes of graphs which have the *diameter-treewidth property* [9], which are precisely apex-minor-free graphs (includes, planar and graphs of bounded genus). For, Smallest Non-Separating Set we show that it does not have polynomial kernel even on planar graphs. We also consider Shortest Non-Separating Path on several classes of graphs. We can observe that the complexity of Shortest Non-Separating Path is closely related to that of Hamiltonian Cycle (or Hamiltonian Path with specified end vertices). This observation readily proves the NP-completeness of Shortest Non-Separating Path on bipartite graphs, split graphs, and planar graphs. For chordal graphs, we devise a polynomial-time algorithm for Shortest Non-Separating Path for the case where k is the shortest path distance between s and t.

Proofs of results for Shortest Non-Separating Path and Smallest Non-Separating Set are similar and the proofs of results for Shortest Non-Disconnecting Path and Smallest Non-Disconnecting Set are similar, in this version of the paper *we only focus on* Shortest Non-Separating Path and Shortest Non-Disconnecting Path.

Related work. The shortest path problem in graphs is one of the most fundamental combinatorial optimization problems, which is studied under various settings. Indeed, our problems Shortest Non-Separating Path and Shortest Non-Disconnecting Path can be seen as variants of this problem. From the computational complexity viewpoint, Shortest Non-Separating Path is known to be NP-hard and its optimization version cannot be approximated with factor $|V|^{1-\varepsilon}$ in polynomial time for $\varepsilon > 0$ unless P = NP [24]. Shortest Non-Disconnecting Path is shown to be NP-hard on general graphs and polynomial-time solvable on chordal graphs [19].

2 Preliminaries

We use standard terminologies and known results in matroid theory and parameterized complexity theory, which are briefly discussed in this section. See [6, 21] for details.

Graphs. Let G be a graph. The vertex set and edge set of G are denoted by $V(G)$ and $E(G)$, respectively. For $v \in V(G)$, the open neighborhood of v in G is denoted by $N_G(v)$ (i.e., $N_G(v) = \{u \in V(G) : \{u, v\} \in E(G)\}$) and the closed neighborhood of v in G is denoted by $N_G[v]$ (i.e., $N_G[v] = N_G(v) \cup \{v\}$). We extend this notation to sets: $N_G(X) = \bigcup_{v \in X} N_G(v) \setminus X$ and $N_G[X] = N_G(X) \cup X$ for $X \subseteq V(G)$. For $u, v \in V(G)$, we denote by $\mathrm{dist}_G(u, v)$ the length of a shortest path between u and v in G, where the length of a path is defined as the number of edges in it. We may omit the subscript of G from these notations when no confusion arises. For $X \subseteq V(G)$, we write $G[X]$ to denote the subgraph of G induced by X. For notational convenience, we may use $G - X$ instead of $G[V(G) \setminus X]$. For $F \subseteq E$, we also use $G - F$ to represent the subgraph of G consisting all vertices of G and all edges in $E \setminus F$. For vertices u and v, a path between u and v is called a *u-v path.* A vertex is called a *pendant* if its degree is exactly 1.

Matroids and representative sets. Let E be a finite set. If $\mathcal{I} \subseteq 2^E$ satisfies the following axioms, the pair $\mathcal{M} = (E, \mathcal{I})$ is called a *matroid*: (1) $\emptyset \in \mathcal{I}$; (2) $Y \in \mathcal{I}$ implies $X \in \mathcal{I}$ for $X \subseteq Y$; and (3) for $X, Y \in \mathcal{I}$ with $|X| < |Y|$, there is $e \in Y \setminus X$ such that $X \cup \{e\} \in \mathcal{I}$. Each set in $\mathcal{I}$ is called an *independent set* of $\mathcal{M}$. From the third axiom of matroids, it is easy to observe that every (inclusion-wise) maximal independent set of $\mathcal{M}$ have the same cardinality. The *rank* of $\mathcal{M}$ is the maximum cardinality of an independent set of $\mathcal{M}$. A matroid $\mathcal{M} = (E, \mathcal{I})$ of rank n is *linear* (or *representable*) over a field $\mathbb{F}$ if there is a matrix $M \in \mathbb{F}^{n \times |E|}$ whose columns are indexed by E such that $X \in \mathcal{I}$ if and only if the set of columns indexed by X is linearly independent in M.

Let $G = (V, E)$ be a graph. A *cographic matroid* of G is a matroid $\mathcal{M}(G) = (E, \mathcal{I})$ such that $F \subseteq E$ is an independent set of $\mathcal{M}(G)$ if and only if $G - F$ is connected. Every cographic matroid is linear and its representation can be computed in polynomial time [21]. Our algorithmic result for Shortest Non-Disconnected Path is based on *representative families* due to [11].

▶ **Definition 3.** *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid and let $\mathcal{F} \subseteq \mathcal{I}$ be a family of independent sets of $\mathcal{M}$. For an integer $q \geq 0$, we say that $\widehat{\mathcal{F}} \subseteq \mathcal{F}$ is q-*representative for $\mathcal{F}$* if the following condition holds: For every $Y \subseteq E$ of size at most q, if there is $X \in \mathcal{F}$ with $X \cap Y = \emptyset$ such that $X \cup Y \in \mathcal{I}$, then there is $\widehat{X} \in \widehat{\mathcal{F}}$ with $\widehat{X} \cap Y = \emptyset$ such that $\widehat{X} \cup Y \in \mathcal{I}$.*

▶ **Theorem 4** ([11, 18]). *Given a linear matroid $\mathcal{M} = (E, \mathcal{I})$ of rank n that is represented as a matrix $M \in \mathbb{F}^{n \times |E|}$ for some field $\mathbb{F}$, a family $\mathcal{F} \subseteq \mathcal{I}$ of independent sets of size p, and an integer q with $p + q \leq n$, there is a deterministic algorithm computing a q-representative family $\widehat{\mathcal{F}} \subseteq \mathcal{F}$ of size $np\binom{p+q}{p}$ with*

$$O\left(|\mathcal{F}| \cdot \left(\binom{p+q}{p} p^3 n^2 + \binom{p+q}{q}^{\omega - 1} \cdot (pn)^{\omega - 1}\right)\right) + (n + |E|)^{O(1)}$$

field operations, where $\omega < 2.373$ is the exponent of the matrix multiplication.

Parameterized complexity. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. A *kernelization* for L is a polynomial-time algorithm that given an instance $(I, k) \in \Sigma^* \times \mathbb{N}$, computes an "equivalent" instance $(I', k') \in \Sigma^* \times \mathbb{N}$ such that (1) $(I, k) \in L$ if and only if $(I', k') \in L$ and (2) $|I'| + k' \leq g(k)$ for some computable function g. We call (I', k') a *kernel.* If the function g is a polynomial, then the kernelization algorithm is called a *polynomial kernelization* and its output (I', k') is called a *polynomial kernel.* An *OR-composition* is an algorithm that given p instances $(I_1, k), \ldots (I_p, k) \in \Sigma^* \times \mathbb{N}$ of L, computes an instance $(I', k') \in \Sigma^* \times \mathbb{N}$ in time $(\sum_{1 \leq i \leq p} |I_i| + k)^{O(1)}$ such that (1) $(I', k') \in L$ if and only if $(I_i, k) \in L$ for some $1 \leq i \leq p$ and (2) $k' = k^{O(1)}$. For a parameterized problem L, its *unparameterized problem* is a language $L' = \{x\#1^k : (x, k) \in L\}$, where $\# \notin \Sigma$ is a blank symbol and $1 \in \Sigma$ is an arbitrary symbol.

▶ **Theorem 5** ([3]). *If a parameterized problem L admits an OR-composition and its unparameterized version is NP-complete, then L does not have a polynomial kernelization unless* $\mathrm{coNP} \subseteq \mathrm{NP/poly}$.

3 Shortest Non-Separating Path

We discuss our complexity and algorithmic results for SHORTEST NON-SEPARATING PATH.

3.1 Hardness on graph classes

We obverse that, in most cases, SHORTEST NON-SEPARATING PATH is NP-hard on classes of graphs for which HAMILTONIAN PATH (with distinguished end vertices) is NP-hard. Let $G = (V, E)$ be a graph and $s, t \in V$ be distinct vertices of G. We add a pendant vertex p adjacent to s and denote the resulting graph by G'. Then, we have the following observation.

▶ **Observation 6.** *For every non-separating path P between s and t in G', $V(G) \setminus V(P) = \{p\}$.*

Suppose that for a class $\mathcal{C}$ of graphs,

- the problem of deciding whether given graph $G \in \mathcal{C}$ has a Hamiltonian path between specified vertices s and t in G is NP-hard and
- $G \in \mathcal{C}$ implies $G' \in \mathcal{C}$.

By Observation 6, G' has a non-separating s-t path if and only if G has a Hamiltonian path between s and t. This implies that the problem of finding a non-separating path between specified vertices is NP-hard on class $\mathcal{C}$.

▶ **Theorem 7.** *The problem of deciding if an input graph has a non-separating s-t path is NP-complete even on planar graphs, bipartite graphs, and split graphs.*

The classes of planar graphs and bipartite graphs are closed under the operation of adding a pendant. Recall that a graph G is a *split graph* if the vertex set $V(G)$ can be partitioned into a clique C and an independent set I. Thus, for the class of split graphs, we need the assumption that the pendant added is adjacent to a vertex in C.

As the problem trivially belongs to NP, it suffices to show that HAMILTONIAN PATH (with distinguished end vertices) is NP-hard on these classes of graphs[1]. For split graphs, it is known that HAMILTONIAN PATH is NP-hard even if the distinguished end vertices are contained in the clique C [20]. Let G be a graph and let $v \in V(G)$. We add a vertex v' that is adjacent to every vertex in $N_G(v)$, that is, v and v' are (false) twins. The resulting graph is denoted by G'. It is easy to verify that G has a Hamiltonian cycle if and only if G' has a Hamiltonian path between v and v'. The class of bipartite graphs is closed under this operation, that is, G' is bipartite. For planar graphs, G' may not be planar in general. However, HAMILTONIAN CYCLE is NP-complete even if the input graph is planar and has a vertex of degree 2 [14]. We apply the above operation to this degree-2 vertex, and the resulting graph G' is still planar. As the problem of finding a Hamiltonian cycle is NP-hard even on bipartite graphs [20] and planar graphs [14], Theorem 7 follows.

[1] These results for bipartite graphs and planar graphs seem to be folklore but we were not able to find particular references.

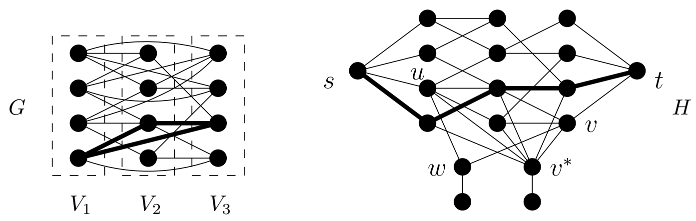

Figure 1 The left figure depicts an instance G of Multicolored Clique and the right figure depicts the graph H constructed from G. Some vertices and edges in H are not drawn in this figure for visibility. The edges of a clique C and the corresponding non-separating s-t path P are drawn as thick lines.

3.2 W[1]-hardness

Next, we show that Shortest Non-Separating Path is W[1]-hard parameterized by k. The proof is done by giving a reduction from Multicolored Clique, which is known to be W[1]-complete [10]. In Multicolored Clique, we are given a graph G with a partition $\{V_1, V_2, \ldots, V_k\}$ of $V(G)$ and asked to determine whether G has a clique C such that $|V_i \cap C| = 1$ for each $1 \leq i \leq k$.

From an instance $(G, \{V_1, \ldots, V_k\})$ of Multicolored Clique, we construct an instance of Shortest Non-Separating Path as follows. Without loss of generality, we assume that G contains more than k vertices. We add two vertices s and t, make s adjacent to all $v \in V_1$ and make t adjacent to all $v \in V_k$. For any pair of $u \in V_i$ and $v \in V_j$ with $|i - j| \geq 2$, we do the following. If $\{u, v\} \in E$, then we remove it. Otherwise, we add a path $P_{u,v}$ of length 2 and a pendant vertex that is adjacent to the internal vertex w of $P_{u,v}$. Finally, we add a vertex v^*, add an edge between v^* and each original vertex $v \in V(G)$, and add a pendant vertex p adjacent to v^*. The constructed graph is denoted by H. See Figure 1 for an illustration of the graph H.

▶ **Lemma 8.** *There is a clique C in G such that $|C \cap V_i| = 1$ for all $1 \leq i \leq k$ if and only if there is a non-separating s-t path of length at most $k + 1$ in H.*

Proof. Suppose first that G has a clique C with $C \cap V_i = \{v_i\}$ for all $1 \leq i \leq k$. Then, $P = \langle s, v_1, v_2, \ldots, v_k, t\rangle$ is an s-t path of length $k + 1$ in H. To see the connectivity of $H - V(P)$, it suffices to show that every vertex is reachable from v^* in $H - V(P)$. By the construction of H, each vertex in $V(G) \setminus V(P)$ is adjacent to v^* in $H - V(P)$. Each vertex z in $V(H) \setminus (V(G) \cup \{v^*, p\})$ is either the internal vertex w of $P_{u,v}$ for some $u, v \in V(G)$ or the pendant vertex adjacent to w. In both cases, at least one of u and v is not contained in P as $V(P) \setminus \{s, t\}$ is a clique in G, implying that z is reachable to v^*.

Conversely, suppose that H has a non-separating s-t path P of length at most $k + 1$ in H. By the assumption that G has more than k vertices, there is a vertex of G that does not belong to P. Observe that P does not contain any internal vertex w of some $P_{u,v}$ as otherwise the pendant vertex adjacent to w becomes an isolated vertex by deleting $V(P)$, which implies $H - V(P)$ has at least two connected components. Similarly, P does not contain v^*. These facts imply that the internal vertices of P belong to $V(G)$, and we have $|V(P) \cap V_i| = 1$ for all $1 \leq i \leq k$. Let $C = V(P) \setminus \{s, t\}$. We claim that C is a clique in G. Suppose otherwise. There is a pair of vertices $u, v \in C$ that are not adjacent in G. This implies that H contains the path $P_{u,v}$. However, as P contains both u and v, the internal vertex of $P_{u,v}$ together with its pendant vertex forms a component in $H - V(P)$, yielding a contradiction that P is a non-separating path in H. ◀

Thus, we have the following theorem.

▶ **Theorem 9.** *Shortest Non-Separating Path is W[1]-hard parameterized by k.*

3.3 Graphs with the diameter-treewidth property

By Theorem 9, Shortest Non-Separating Path is unlikely to be fixed-parameter tractable on general graphs. To overcome this intractability, we focus on sparse graph classes. We first note that the algorithmic meta-theorems for FO Model Checking [12, 13] do not seem to be applicable to Shortest Non-Separating Path as we need to care about the connectivity of graphs. However, the property that vertex set X forms a non-separating s-t path can be expressed as:

$$\texttt{conn}(V \setminus X) \wedge \texttt{hampath}(X, s, t),$$

where $\texttt{conn}(Y)$ and $\texttt{hampath}(Y, s, t)$ are formulas in MSO_2 such that $\texttt{conn}(Y)$ (resp. $\texttt{hampath}(Y, s, t)$) is true if and only of the subgraph induced by Y is connected and (resp. the subgraph induced by Y has a Hamiltonian path between s and t). We omit the details of these formulas, which can be found in [6] for example[2]. By Courcelle's theorem [5] and its extension due to Arnborg et al. [1], we can compute a shortest non-separating s-t path in $O(f(\mathrm{tw}(G))n)$ time, where n is the number of vertices and $\mathrm{tw}(G)$ is the treewidth[3] of G. As there is an $O(\mathrm{tw}(G)^{\mathrm{tw}(G)^3}n)$-time algorithm for computing the treewidth of an input graph G [2], we have the following theorem.

▶ **Theorem 10.** *Shortest Non-Separating Path is fixed-parameter tractable parameterized by the treewidth of input graphs.*

A class $\mathcal{C}$ of graphs is *minor-closed* if every minor of a graph $G \in \mathcal{C}$ also belongs to $\mathcal{C}$. We say that $\mathcal{C}$ has the *diameter-treewidth property* if there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that for every $G \in \mathcal{C}$, the treewidth of G is upper bounded by $f(\mathrm{diam}(G))$, where $\mathrm{diam}(G)$ is the diameter of G. It is well known that every planar graph G has treewidth at most $3 \cdot \mathrm{diam}(G) + 1$ [22][4], which implies that the class of planar graphs has the diameter-treewidth property. This can be generalized to more wider classes of graphs. A graph is called an *apex graph* if it has a vertex such that removing it makes the graph planar.

▶ **Theorem 11** ([7, 9]). *Let $\mathcal{C}$ be a minor-closed class of graphs. Then, $\mathcal{C}$ has the diameter-treewidth property if and only if it excludes some apex graph.*

For $C \subseteq V(G)$ that induces a connected subgraph $G[C]$, we denote by G_C the graph obtained from G by contracting $G[C]$ into a single vertex v_C and making v_C adjacent to all the vertices in $N(C)$. Since $G[C]$ is connected, vertex v_C is well-defined.

▶ **Lemma 12.** *Let $C \subseteq V(G)$ be a vertex subset such that $G[C]$ is connected. Let P be an s-t path in G with $V(P) \cap C = \emptyset$. Then, P is non-separating in G if and only if it is non-separating in G_C.*

---

[2] In [6], they give an MSO_2 sentence $\texttt{hamiltonicity}$ expressing the property of having a Hamiltonian cycle, which can be easily transformed into a formula expressing $\texttt{hampath}(X, s, t)$.

[3] We do not give the definition of treewidth and (the optimization version of) Courcelle's theorem. We refer to [6] for details.

[4] More precisely, the treewidth of a planar graph is upper bounded by $3r + 1$, where r is the radius of the graph.

Proof. Suppose first that P is non-separating in G. Let $u, v \in V(G) \setminus V(P)$ be arbitrary. As P is non-separating, there is a u-v path P' in $G - V(P)$. Let u' be the vertex of G_C such that $u' = u$ if $u \notin C$ and $u' = v_C$ if $u \in C$. Let v' be the vertex defined analogously. We show that there is a u'-v' path in $G_C - V(P)$ as well. If P' does not contain any vertex in C, then it is also a u'-v' path in G_C, and hence we are done. Suppose otherwise. Let x and y be the vertices in $V(P') \cap C$ that are closest to u and v, respectively. Note that x and y can be the end vertices of P', that is, C may contain u and v. Let $P_{u,x}$ and (resp. $P_{y,v}$) be the subpath of P' between u and x (resp. y and v). Then, the sequence of vertices obtained by concatenating $P_{u,x}$ after $P_{y,v} - \{y\}$ and replacing exactly one occurrence of a vertex in C with v_C forms a path between u' and v'. Since we choose u, v arbitrarily, there is a path between any pair of vertices in $G_C - V(P)$ as well. Hence, P is non-separating in G_C.

Conversely, suppose that P is non-separating in G_C. For $u, v \in V(G_C) \setminus V(P)$, there is a path P' in $G_C - V(P)$. Suppose that neither $u = v_C$ nor $v = v_C$. Then, we can construct a u-v path in $G - V(P)$ as follows. If $v_C \notin V(P')$, P' is also a path in $G - V(P)$ and hence we are done. Otherwise, $v_C \in V(P')$. Let P_u and P_v be the subpaths in $P' - \{v_C\}$ containing u and v, respectively. From P_u and P_v, we have a u-v path in G by connecting them with an arbitrary path in $G[C]$ between the end vertices other than u and v. Note that such a bridging path in $G[C]$ always exists since $G[C]$ is connected. Moreover, as $V(P') \cap C = \emptyset$ and $V(P) \cap C = \emptyset$, this is also a u-v path in $G - V(P)$. Suppose otherwise that either $u = v_C$ or $v = v_C$, say $u = v_C$. In this case, we can construct a path between every vertex w in C and v by concatenating P' and an arbitrary path in $G[C]$ between w and the end vertex of the subpath $P' - \{v_C\}$ other than v. Therefore, P is non-separating in G. ◀

Now, we are ready to state the main result of this subsection.

▶ **Theorem 13.** *Let $\mathcal{C}$ be a graph class excluding a fixed apex graph H as a minor. Then,* Shortest Non-Separating Path *on $\mathcal{C}$ is fixed-parameter tractable parameterized by k.*

Proof. Let $G \in \mathcal{C}$. We first compute $B = \{v \in V(G) : \mathrm{dist}(s, v) \le k\}$. This can be done in linear time. If $t \notin B$, then the instance (G, s, t, k) is trivially infeasible. Suppose otherwise that $t \in B$. Let C be a component in $G - B$. By definition, every non-separating s-t path P of length at most k does not contain any vertex of C. Let G' be the graph obtained from G by contracting all edges in $E(G - B)$. For each component C in $G - B$, we denote by v_C the vertex of G' corresponding to C (i.e., v_C is the vertex obtained by contracting all edges in C). Then, we have $\mathrm{diam}(G') \le 2k + 2$ as $\mathrm{diam}(G[B]) \le k$ and every vertex in $V(G') \setminus B$ is adjacent to a vertex in B. By Lemma 12, G has a non-separating s-t path of length at most k if and only if so does G'. Since $\mathcal{C}$ is minor-closed, we have $G' \in \mathcal{C}$ and hence the treewidth of G' is upper bounded by $f(2k + 2)$ for some function f. By Theorem 10, we can check whether G' has a non-separating s-t path of length at most k in $O(g(k)|V(G')|)$ time. ◀

Theorem 10 does not give precise dependence on $\mathrm{tw}(G)$ in the running time of the algorithm. In fact, given a tree decomposition of G of width $\mathrm{tw}(G)$ we can design an algorithm for Shortest Non-Separating Path running in time $2^{O(\mathrm{tw}(G))} n^{O(1)}$, using matroid based tools [11]. Further, there exists a factor-2 approximation for $\mathrm{tw}(G)$ running in time $2^{O(\mathrm{tw}(G))} n$ [16]. Combined with this we get the following result.

▶ **Theorem 14.** *There exists an algorithm for Shortest Non-Separating Path running in time $2^{O(\mathrm{tw}(G))} n^{O(1)}$.*

The proof of this result is based on the standard dynamic programming over graphs of bounded treewidth together with representative sets and will appear in the final version of the paper. Applying the result of Theorem 14 in Theorem 13, we get the following results.

▶ **Theorem 15.** *Let $\mathcal{C}$ be a graph class excluding a fixed apex graph H as a minor. Then,* Shortest Non-Separating Path *on $\mathcal{C}$ admits an algorithm with running time $2^{O(k)}n^{O(1)}$.*

3.4 TI-Shortest Non-Separating Path on Unit Disk Graphs

Given n unit disks in the plane, a unit disk graph G consists of n vertices, corresponding to each disk, and there is an edge between two vertices if and only if the corresponding unit disks intersect. For our problem, we are given a unit disk graph $G = (V, E)$ and its representation $(\mathcal{D}, C)$. Let $V = \{v_1, v_2, \cdots, v_n\}$ and for each v_i, D_i be the corresponding disk centered at c_i. Here $\mathcal{D} = \{D_i | \ 1 \leq i \leq n\}$ and $C = \{c_i | \ 1 \leq i \leq n\}$.

▶ **Reduction Rule 1.** *If G has more than two connected components then return* NO.

▶ **Lemma 16.** *Reduction Rule 1 is safe and can be implemented in polynomial time.*

Let us assume that the graph has exactly two connected components. Observe that the given instance is a YES instance if and only if at least one of the two components contains exactly k vertices. Otherwise, the instance is a NO instance. The number of components in the given graph and the number of vertices in each component can be checked in polynomial time. Thus now onward we assume that G is connected. Let us consider a $(\frac{1}{2} \times \frac{1}{2})$ square grid on the plane. Let V_S be the set of centers of the disks that are contained inside a grid cell S; more formally, $V_S = \{v_i | \ c_i \in S\}$. We also define $N(V', S)$ as the set of vertices in the cell S that are neighbors of vertices in V'. For any cell S in the grid, next, we prove that if S contains at least $k + 24$ centers then the given instance is a YES instance.

▶ **Reduction Rule 2.** *If there exists a cell S with at least $k + 24$ centers, return YES.*

▶ **Lemma 17.** *Reduction Rule 2 is safe and can be implemented in polynomial time.*

From now onwards we assume that each cell has at most $k + 23$ centers.

▶ **Theorem 18.** TI-Shortest Non-Separating Path *can be solved in time $2^{O(k \log k)}n^{O(1)}$ on unit disk graphs.*

Proof. We guess the first vertex of the path, say v_i. Let Y be the set of vertices in the circle drawn centering c_i with radius k. Observe that the vertices of the solution path P must be a subset of Y. There are at most $\mathcal{O}(k^2)$ cells inside the circle drawn centering c_i with radius k. By Reduction Rule 2, no cell contains more than $k + 23$ vertices. Thus Y contains at most $O(k^3)$ vertices. Now all we need to do is to guess a subset X of size k as potential vertices of the path P and test that indeed it forms a path and $G - X$ is connected. All this can be done in $\binom{O(k^3)}{k} \times k! \times n^{O(1)} = 2^{O(k \log k)}n^{O(1)}$ time, concluding the proof. ◀

3.5 Chordal graphs with $k = \text{dist}(s, t)$

In Section 3.1, we have seen that Shortest Non-Separating Path is NP-complete even on split graphs (and thus on more general chordal graphs as well). To overcome this intractability, we restrict ourselves to finding a non-separating s-t path of length $\text{dist}(s, t)$ on chordal graphs.

A graph G is *chordal* if it has no cycles of length at least 4 as an induced subgraph. In the following, we fix a connected chordal graph G.

▶ **Lemma 19.** *Let $S \subseteq V(G)$ be a vertex set such that $G[S]$ is connected. For $u, v \in S$, every induced u-v path P in G satisfies that $V(P) \subseteq N[S]$.*

For $u, v \in V(G)$, a set of vertices $S \subseteq V(G) \setminus \{u, v\}$ is called a *u-v separator* of G if there is no u-v path in $G - S$. An inclusion-wise minimal u-v separator of G is called a *minimal u-v separator.* A *minimal separator* of G is a minimal u-v separator for some $u, v \in G$. Dirac's well-know characterization [8] of chordal graphs states that a graph is chordal if and only if every minimal separator induces a clique.

▶ **Lemma 20.** *Let $s, t \in V(G)$ be such that $\{s, t\} \notin E(G)$. If $v \in V(G) \setminus \{s, t\}$ is an internal vertex of a shortest s-t path P, then $N[v] \setminus \{s, t\}$ is an s-t separator of G.*

Proof. Let $d = \mathrm{dist}(s, t)$. For $0 \le i \le d$, let

$$D_i = \{v \in V(G) : \mathrm{dist}(s, v) = i \wedge \mathrm{dist}(v, t) = d - i\}$$

and $V(P) \cap D_i = \{u_i\}$. Observe that each D_i is a clique: if $i \in \{0, d\}$, then it is a singleton; otherwise, it is a minimal s-t separator of the chordal graph $G[\bigcup_{0\le j\le d} D_j]$, meaning that D_i is a clique. From this observation, we have $D_i \subseteq N[u_i] \setminus \{s, t\}$ for $0 < i < d$. Let j ($0 < j < d$) be the index such that $v = u_j$.

Suppose to the contrary that there is an induced s-t path Q such that $V(Q) \cap (N[u_j] \setminus \{s, t\}) = \emptyset$. By Lemma 19, $V(Q) \subseteq N[V(P)] = \bigcup_{0\le i\le d} N[u_i]$ holds. Since Q starts in $N[u_0]$ and ends in $N[u_d]$, there are indices i and k with $0 \le i < j < k \le d$ such that Q consecutively visits a vertex $v_i \in N[u_i]$ and then a vertex $v_k \in N[u_k]$ in this order. Since $\mathrm{dist}(u_i, u_k) = k - i \ge 2$ and $\{v_i, v_k\} \in E$, at least one of $v_i \ne u_i$ and $v_k \ne u_k$ holds. By symmetry, we assume that $v_i \ne u_i$.

If $v_k = u_k$, then $v_i \in N(u_i) \cap N(u_k)$. In this case, we have $i = j - 1$ and $k = j + 1$ since otherwise P admits a shortcut using the subpath $\langle u_i, v_i, u_k\rangle$. This implies that $\mathrm{dist}(s, v_i) \le \mathrm{dist}(s, u_i) + 1 = i + 1 = j$ and $\mathrm{dist}(v_i, t) \le 1 + \mathrm{dist}(v_k, t) = 1 + \mathrm{dist}(u_k, t) = 1 + (d - k) = d - j$. Since $\mathrm{dist}(s, v_i) + \mathrm{dist}(v_i, t) \ge d$, we have $\mathrm{dist}(s, v_i) = j$ and $\mathrm{dist}(v_i, t) = d - j$. This implies that $v_i \in D_j \subseteq N[u_j] \setminus \{s, t\}$, a contradiction.

Next we consider the case $v_k \ne u_k$. Recall that we also have $v_i \ne u_i$ as an assumption. In this case, we have $k - i \le 3$ as $\langle u_i, v_i, v_k, u_k\rangle$ is not a shortcut for P. Assume first that $k - i = 3$. By symmetry, we may assume that $i = j - 1$ and $k = j + 2$. Since $\mathrm{dist}(s, v_i) \le \mathrm{dist}(s, u_i) + 1 = j$ and $\mathrm{dist}(v_i, t) \le 2 + \mathrm{dist}(u_k, t) \le 2 + (d - k) = d - j$, we have $v_i \in D_j \subseteq N[u_j] \setminus \{s, t\}$, a contradiction. Next assume that $k - i = 2$. That is, $i = j - 1$ and $k = j + 1$. Since $v_i, v_k \notin N[u_j] \setminus \{s, t\}$ and P is shortest, the vertices v_i, u_i, u_j, u_k, v_k are distinct and form a cycle of length 5. Observe that $v_i \notin \{s, t\}$ since otherwise $\langle v_i = s, v_k, u_k\rangle$ or $\langle u_i, v_i = t\rangle$ is a shortcut. Similarly, $v_k \notin \{s, t\}$. Hence, $v_i, v_k \notin N[u_j]$. Therefore, the possible chords for the cycle $\langle v_i, u_i, u_j, u_k, v_k\rangle$ are $\{u_i, v_k\}$ and $\{u_k, v_i\}$. In any combination of them, the graph has an induced cycle of length at least 4. ◀

Let d and D_i be defined as in the proof of Lemma 20, and let $D = \bigcup_{0\le i\le d} D_i$. Recall that each D_i is a clique. Observe that if $|D_i| = 1$ for all $0 \le i \le d$, then G contains a unique shortest s-t path, and thus the problem is trivial. Otherwise, we define ℓ to be the minimum index such that $|D_\ell| > 1$ and r to be the maximum index such that $|D_r| > 1$. Since $|D_0| = |D_d| = 1$, we have $0 < \ell \le r < d$.

Our algorithm works as follows.

1. If G contains a unique shortest s-t path P, then test if P is non-separating.
2. Otherwise, find a shortest s-t path P satisfying the following conditions.
   a. $V(P)$ does not contain a minimal a-b separator for all $a \in D_\ell$ and $b \in V \setminus D$.
   b. $V(P)$ does not contain a minimal a-b separator for all $a \in D_\ell$ and $b \in D_r$.

▶ **Lemma 21.** *The algorithm is correct.*

▶ **Lemma 22.** *The algorithm has a polynomial-time implementation.*

We do not optimize the running time of the above algorithm, and a straightforward implementation runs in time $O(n^2m)$, where $n = |V(G)|$ and $m = |E(G)|$, which might be improved with some data structure.

▶ **Theorem 23.** *There is a polynomial-time algorithm for* Shortest Non-Separating Path *on chordal graphs, given that k is equal to the shortest path distance between s and t.*

4 Shortest Non-Disconnecting Path

The goal of this section is to establish the fixed-parameter tractability and a conditional lower bound on polynomial kernelizations for Shortest Non-Disconnecting Path.

4.1 Fixed-parameter tractability

▶ **Theorem 24.** Shortest Non-Disconnecting Path *can be solved in time $2^{\omega k} n^{O(1)}$, where ω is the matrix multiplication exponent and n is the number of vertices of G.*

To prove this theorem, we give a dynamic programming algorithm with the aid of representative families of cographic matroids. Let (G, s, t, k) be an instance of Shortest Non-Disconnecting Path. For $0 \le i \le k$ and $v \in V(G)$, we define $\mathsf{dp}(i, v)$ as the family of all sets of edges F satisfying the following two conditions: (1) F is the set of edges of an s-v path of length i and (2) $G - F$ is connected. An edge set F is *legitimate* if F forms a path and $G - F$ is connected. For a family of edge sets $\mathcal{F}$ and an edge e, we define $\mathcal{F} \bowtie e := \{F \cup \{e\} : F \in \mathcal{F}\}$ and $\mathsf{leg}(\mathcal{F})$ as the subfamily of $\mathcal{F}$ consisting of all legitimate $F \in \mathcal{F}$. The following simple recurrence correctly computes $\mathsf{dp}(i, v)$.

$$\mathsf{dp}(i,v) = \begin{cases} \{\emptyset\} & i = 0 \text{ and } s = v \quad (3) \\ \emptyset & i = 0 \text{ and } s \neq v \quad (4) \\ \mathsf{leg}\left(\bigcup_{u \in N(v)} (\mathsf{dp}(i-1, u) \bowtie \{u, v\}) \right) & i > 0\ . \quad (5) \end{cases}$$

A straightforward induction proves that $\mathsf{dp}(i, t) \neq \emptyset$ if and only if G has a non-disconnecting s-t path of length exactly i and hence it suffices to check whether $\mathsf{dp}(i, t) \neq \emptyset$ for $0 \le i \le k$. However, the running time to evaluate this recurrence is $n^{O(k)}$. To reduce the running time of this algorithm, we apply Theorem 4 to each $\mathsf{dp}(i, v)$. Now, instead of (5), we define

$$\mathsf{dp}(i,v) = \mathsf{rep}_{k-i}\left(\mathsf{leg}\left(\bigcup_{u \in N(v)} (\mathsf{dp}(i-1, u) \bowtie \{u, v\}) \right) \right), \quad (3')$$

where $\mathsf{rep}_{k-i}(\mathcal{F})$ is a $(k - i)$-representative family of $\mathcal{F}$ for the cographic matroid $\mathcal{M} = (E(G), \mathcal{I})$ defined on G. In the following, we abuse the notation of dp to denote the families of legitimate sets that are computed by the recurrence composed of (3), (4), and (3').

▶ **Lemma 25.** *The recurrence composed of (3), (4), and (3') is correct, that is, G has a non-disconnecting s-t path of length at most k if and only if $\bigcup_{0 \le i \le k} \mathsf{dp}(i, t) \neq \emptyset$.*

Proof. It suffices to show that $\mathsf{dp}(k', t) \neq \emptyset$ if G has a non-disconnecting s-t path P of length $k' \leq k$. Let $P = (v_0 = s, v_1, \ldots, v_{k'} = t)$ be a non-disconnecting path in G. We assume that G has no non-disconnecting s-t path of length strictly smaller than k'. For $0 \leq i \leq k'$, we let $P_i = (v_i, v_{i+1}, \ldots, v_{k'})$. In the following, we prove, by induction on i, a slightly stronger claim that there is a legitimate set $F \in \mathsf{dp}(i, v_i)$ such that $F \cup E(P_i)$ forms a non-disconnecting s-t path in G for all $0 \leq i \leq k'$. As $\mathsf{dp}(0, s) = \{\emptyset\}$ and $P_0 = P$ itself is a non-disconnecting path, we are done for $i = 0$. Suppose that $i > 0$. By the induction hypothesis, there is a legitimate $F \in \mathsf{dp}(i-1, v_{i-1})$ such that $F \cup E(P_{i-1})$ forms a non-disconnecting s-t path in G. Note that $F \cap E(P_{i-1}) = \emptyset$ as otherwise G has a non-disconnecting s-t path of length smaller than k'. Let $\mathcal{F} = \mathsf{leg}(\bigcup_{u \in N(v_i)} (\mathsf{dp}(i-1) \bowtie \{u, v_i\}))$. Since $F \cup E(P_{i-1})$ is legitimate, $F \cup \{\{v_{i-1}, v_i\}\}$ is also legitimate, implying that $\mathcal{F}$ is nonempty. Let $\widehat{\mathcal{F}} = \mathsf{rep}_{k-i}(\mathcal{F})$ be $(k-i)$-representative for $\mathcal{F}$, $X = F \cup \{\{v_{i-1}, v_i\}\}$, and let $Y = E(P_i)$. As $|Y| \leq k - i$, $X \cap Y = \emptyset$, and $X \cup Y \in \mathcal{I}$, $\widehat{\mathcal{F}}$ contains an edge set $\widehat{X}$ with $\widehat{X} \cap Y = \emptyset$ and $\widehat{X} \cup Y \in \mathcal{I}$, implying that there is $\widehat{X} \in \mathsf{dp}(i, v_i)$ such that $\widehat{X} \cup E(P_i)$ forms a non-disconnecting s-t path in G. ◀

▶ **Lemma 26.** *The recurrence can be evaluated in time $2^{\omega k} n^{O(1)} \subset 5.18^k n^{O(1)}$, where $\omega < 2.373$ is the exponent of the matrix multiplication.*

Proof. By Theorem 4, $\mathsf{dp}(i, v)$ contains at most $2^k kn$ sets for $0 \leq i \leq k$ and $v \in V(G)$ and can be computed in time $2^{\omega k} n^{O(1)}$. ◀

Thus, Theorem 24 follows.

4.2 Kernel lower bound

It is well known that a parameterized problem is fixed-parameter tractable if and only if it admits a kernel (see [6], for example). By Theorem 24, Shortest Non-Disconnecting Path admits a kernel. A next natural step next is to explore the existence of polynomial kernel for Shortest Non-Disconnecting Path. However, the following theorem conditionally rules out the possibility of polynomial kernelization. We first show the following lemma.

▶ **Lemma 27.** *Let H be a connected graph. Suppose that H has a cut vertex v. Let C be a component in $H - \{v\}$ and let $F \subseteq E(H[C \cup \{v\}])$. Then, $H - F$ is connected if and only if $H[C \cup \{v\}] - F$ is connected.*

Proof. If $H - F$ is connected, then all the vertices in $C \cup \{v\}$ are reachable from v in $H - F$ without passing through any vertex in $V(H) \setminus (\{C\} \cup \{v\})$. Thus, such vertices are reachable from v in $H[C \cup \{v\}] - F$. Conversely, suppose $H[C \cup \{v\}] - F$ is connected. Then, every vertex in C is reachable from v in $H - F$. Moreover, as F does not contain any edge outside $H[C \cup \{v\}]$, every other vertex is reachable from v in $H - F$ as well. ◀

▶ **Theorem 28.** *Unless* coNP $\subseteq$ NP/poly*, Shortest Non-Disconnecting Path does not admit a polynomial kernelization (with respect to parameter k).*

Proof. We give an OR-composition for Shortest Non-Disconnecting Path. Let $(G_1, s_1, t_1, k), \ldots, (G_p, s_p, t_p, k)$ be p instances of Shortest Non-Disconnecting Path.

We assume that for $1 \leq i \leq p$, t_i is not a cut vertex in G_i. To justify this assumption, suppose that t_i is a cut vertex in G_i. Let C be the component in $G_i - \{t_i\}$ that contains s_i. By Lemma 27, for any s_i-t_i path, it is non-disconnecting in G_i if and only if so is in $G_i[C \cup \{t_i\}]$. Thus, by replacing G_i with $G_i[C]$, we can assume that t_i is not a cut vertex in G_i.

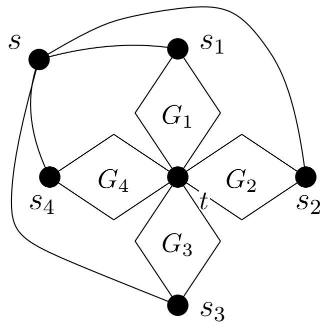

Figure 2 An illustration of the graph G obtained from $q = 4$ instances.

From the disjoint union of $G_1, \ldots, G_p$, we construct a single instance (G, s, t, k') as follows. We first add a vertex s and an edge between s and s_i for each $1 \leq i \leq p$. Then, we identify all t_i's into a single vertex t. See Figure 2 for an illustration.

In the following, we may not distinguish t from t_i. Now, we claim that $(G, s, t, k+1)$ is a yes-instance if and only if (G_i, s_i, t_i, k) is a yes-instance for some i.

Consider an arbitrary s-t path in G. Observe that all edges in the path except for the one incident to s are contained in a single subgraph G_i for some $1 \leq i \leq p$ as $\{s, t\}$ separates $V(G_i) \setminus \{t_i\}$ from $V(G_j) \setminus \{t_j\}$ for $j \neq i$. Moreover, the path P forms $P = (s, s_i, v_1, \ldots, v_q, t)$, meaning that the subpath $P' = (s_1, v_1, \ldots, v_q, t_i)$ is an s_i-t_i path in G_i. This conversion is reversible: for any s_i-t_i path P' in G_i, the path obtained from P' by attaching s adjacent to s_i is an s-t path in G. Thus, it suffices to show that for $F \subseteq E(G_i)$, $F \cup \{\{s, s_i\}\}$ is a cut of G if and only if F is a cut of G_i. Since t is a cut vertex in $G - \{\{s, s_i\}\}$, by Lemma 27, the claim holds. ◀

We obtain the following result for the Smallest Non-Separating Set problem.

▶ **Theorem 29.** *Unless* coNP $\subseteq$ NP/poly, Smallest Non-Separating Set *does not admit a polynomial kernelization (with respect to parameter k) on planar graphs.*

Proof. We give an OR-composition for Smallest Non-Separating Set. Given t planar graphs $G_1, G_2, \cdots, G_t$ where each G_i contains n_i many vertices. We construct a new planar graph G' as follows. We create $k+2$ many copies for each planar graph G_i. For each planar graph G and a vertex $v \in V(G)$ there exists a planar embedding with v on the outer-face. Fix an arbitrary set of $k+2$ *distinct* vertices in G_i, say $v^1, v^2, \cdots, v^{k+2}$. Then, we obtain embeddings for $G_i^1, G_i^2, \cdots, G_i^{k+2}$ such that $v_i^1, v_i^2, \cdots, v_i^{k+2}$ are on the outer-face, respectively. Now add an edge between v_i^ℓ and $v_i^{\ell+1}$ for all $1 \leq \ell \leq k+1$. Also, add an edge between v_i^{k+2} and v_{i+1}^1 for all $1 \leq i \leq t-1$. That is these planar graphs are chained into a path. It is easy to see that the resulting instance G' is planar.

▷ Claim 30. G' is a YES instance if and only if at least one of the G_i is a YES instance.

Proof. We prove the forward direction first. Assume that in G', there exists a connected set X of size k in G' such that $G' - X$ is connected. Notice, deleting any vertex v_i^j from G_i^j where $j \in [k+1]$ makes the graph G' disconnected. Thus X can not contain any vertex v_i^j from G_i^j where $j \in [k+1]$. Hence, we can assume that $X \cap \{v_i^j\} = \emptyset$ in G_j for $1 \leq j \leq k+2$, i.e X is contained completely inside any one of G_i^j without containing any vertex at the outer face that shares edges with other copies of G_i. Therefore, X is of size k inside G_i^j and $G_i^j - X$ is connected. Hence, G_i is a YES instance.

Now, we prove the reverse direction. Assume that there is a graph, G_i in which there exists a connected set X of size k such that $G_i - X$ is connected. Without loss of generality, assume $X = \{v_1, v_2, \cdots, v_k\}$. Since G_i contains X, all the copies of G_i in G' also contain X. Now, we show that there exists a copy of G_i, say G_i^p in G', that does not contain any vertex of X on the outer face which shares edges with other copies of G_i. As $|X| = k$, observe that there can be at most k many different copies of G_i in G' which has a vertex of X on the outer faces which shares edges with the other copies. Hence, by pigeon hole principle, there exists at least a copy of G_i in G' which has no vertex on the outer face that shares edges with other copies. Thus, deleting X from that copy will not disconnect any G_i^a or G_j^b in the graph G'. Therefore, G' is also a YES instance. ◁

This concludes the proof. ◀

References

1 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991. doi:10.1016/0196-6774(91)90006-K.

2 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

3 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. doi:10.1016/j.jcss.2009.04.001.

4 Guantao Chen, Ronald J. Gould, and Xingxing Yu. Graph connectivity after path removal. *Comb.*, 23(2):185–203, 2003. doi:10.1007/s003-0018-z.

5 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edition, 2015.

7 Erik D. Demaine and Mohammad Taghi Hajiaghayi. Diameter and treewidth in minor-closed graph families, revisited. *Algorithmica*, 40(3):211–215, 2004. doi:10.1007/s00453-004-1106-1.

8 G. A. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25:71–76, 1961.

9 David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000. doi:10.1007/s004530010020.

10 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009. doi:10.1016/j.tcs.2008.09.065.

11 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):29:1–29:60, 2016.

12 Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. In *AMS-ASL Joint Special Session*, volume 558 of *Contemporary Mathematics*, pages 181–206. American Mathematical Society, 2009.

13 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095.

14 Alon Itai, Christos H. Papadimitriou, and Jayme Luiz Szwarcfiter. Hamilton paths in grid graphs. *SIAM J. Comput.*, 11(4):676–686, 1982. doi:10.1137/0211056.

15 Ken-ichi Kawarabayashi, Orlando Lee, Bruce A. Reed, and Paul Wollan. A weaker version of lovász' path removal conjecture. *J. Comb. Theory, Ser. B*, 98(5):972–979, 2008. doi:10.1016/j.jctb.2007.11.003.

16 Tuukka Korhonen. A single-exponential time 2-approximation algorithm for treewidth. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 184–192. IEEE, 2021. doi:10.1109/FOCS52979.2021.00026.

17 Matthias Kriesell. Induced paths in 5-connected graphs. *J. Graph Theory*, 36(1):52–58, 2001. doi:10.1002/1097-0118(200101)36:1<52::AID-JGT5>3.0.CO;2-N.

18 Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. Deterministic truncation of linear matroids. *ACM Trans. Algorithms*, 14(2):14:1–14:20, 2018.

19 Xiao Mao. Shortest non-separating st-path on chordal graphs. *CoRR*, abs/2101.03519, 2021. arXiv:2101.03519.

20 Haiko Müller. Hamiltonian circuits in chordal bipartite graphs. *Discret. Math.*, 156(1-3):291–298, 1996.

21 James G. Oxley. *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, Inc., USA, 2006.

22 Neil Robertson and Paul D. Seymour. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.

23 William T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, s3-13(1):743–767, 1963. doi:10.1112/plms/s3-13.1.743.

24 Bang Ye Wu and Hung-Chou Chen. The approximability of the minimum border problem. In *The 26th Workshop on Combinatorial Mathematics and Computation Theory*, 2009.