

Parameterized Maximum Node-Disjoint Paths

Michael Lampis 

Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Manolis Vasilakis 

Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Abstract

We revisit the MAXIMUM NODE-DISJOINT PATHS problem, the natural optimization version of the famous NODE-DISJOINT PATHS problem, where we are given an undirected graph G , k (demand) pairs of vertices (s_i, t_i) , and an integer ℓ , and are asked whether there exist at least ℓ vertex-disjoint paths in G whose endpoints are given pairs. This problem has been intensely studied from both the approximation and parameterized complexity point of view and is notably known to be intractable by standard structural parameters, such as tree-depth, as well as the combined parameter ℓ plus pathwidth. We present several results improving and clarifying this state of the art, with an emphasis towards FPT approximation.

Our main positive contribution is to show that the problem’s intractability can be overcome using approximation: We show that for several of the structural parameters for which the problem is hard, most notably tree-depth, the problem admits an *efficient FPT approximation scheme*, returning a $(1 - \varepsilon)$ -approximate solution in time $f(\text{td}, \varepsilon)n^{\mathcal{O}(1)}$. We manage to obtain these results by comprehensively mapping out the structural parameters for which the problem is FPT if ℓ is also a parameter, hence showing that understanding ℓ as a parameter is key to the problem’s approximability. This, in turn, is a problem we are able to solve via a surprisingly simple color-coding algorithm, which relies on identifying an insightful problem-specific variant of the natural parameter, namely the number of vertices used in the solution.

The results above are quite encouraging, as they indicate that in some situations where the problem does not admit an FPT algorithm, it is still solvable almost to optimality in FPT time. A natural question is whether the FPT approximation algorithm we devised for tree-depth can be extended to pathwidth. We resolve this negatively, showing that under the *Parameterized Inapproximability Hypothesis* no FPT approximation scheme for this parameter is possible, even in time $f(\text{pw}, \varepsilon)n^{g(\varepsilon)}$. We thus precisely determine the parameter border where the problem transitions from “hard but approximable” to “inapproximable”.

Lastly, we strengthen existing lower bounds by replacing W[1]-hardness by XNLP-completeness for parameter pathwidth, and improving the $n^{o(\sqrt{\text{td}})}$ ETH-based lower bound for tree-depth to (the optimal) $n^{o(\text{td})}$.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases ETH, Maximum Node-Disjoint Paths, Parameterized Complexity, PIH

Digital Object Identifier 10.4230/LIPIcs.IPEC.2025.3

Related Version *Full Version*: <https://arxiv.org/abs/2404.14849>

Funding This work is partially supported by ANR project ANR-21-CE48-0022 (S-EX-AP-PE-AL).

1 Introduction

One of the most important problems of structural graph theory has arguably been NODE-DISJOINT PATHS, where given a graph G and k pairs of its vertices (s_i, t_i) for $i = 1, \dots, k$, called demands, the goal is to determine whether there exist k vertex-disjoint paths connecting s_i and t_i . This extensively studied problem [1, 4, 5, 20, 22, 24, 25, 32, 34] is one of the very first to be proven NP-complete (for k being part of the input) [19], and it has a central



© Michael Lampis and Manolis Vasilakis;

licensed under Creative Commons License CC-BY 4.0

20th International Symposium on Parameterized and Exact Computation (IPEC 2025).

Editors: Akanksha Agrawal and Erik Jan van Leeuwen; Article No. 3; pp. 3:1–3:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

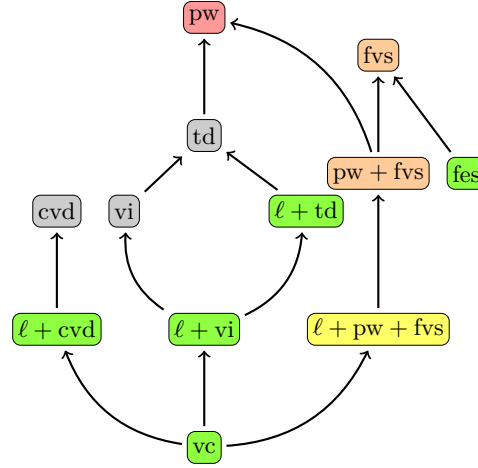
role in the field of structural graph theory as well as in parameterized complexity [27], as the breakthrough result by Robertson and Seymour [31] that it is fixed-parameter tractable (FPT) parameterized by k (i.e., admits an $f(k)n^{\mathcal{O}(1)}$ algorithm for some function f , where $n = |V(G)|$) is the culmination of their long and influential series of works on Graph Minors.

In this work we concern ourselves with MAXIMUM NODE-DISJOINT PATHS (MAXNDP), the natural generalization of NODE-DISJOINT PATHS where one asks whether at least $\ell \leq k$ demands can be routed by vertex-disjoint paths (we say that a demand is routed when there exists a path connecting its endpoints in the set of vertex-disjoint paths of the solution). Notice that one could alternatively phrase this as an optimization problem and ask for the maximum number of demands that can be routed. Even though MAXNDP has been intensely studied with respect to its approximability [7, 8, 9, 10, 21], our understanding regarding its tractability under the perspective of parameterized complexity is rather limited. Given the rich literature regarding NODE-DISJOINT PATHS and the importance of its structural parameterizations (indeed, Scheffler’s $\text{tw}^{\mathcal{O}(\text{tw})}n^{\mathcal{O}(1)}$ algorithm [32] is a key ingredient of the proof of [31]), the quest to study MAXNDP under the same point of view is strongly motivated, with the hope of extending some of these results to it. Alas, prior work by Ene, Mnich, Pilipczuk, and Risteski [14] shows that MAXNDP is already W[1]-hard when parameterized by the tree-depth of the input graph (in fact their proof implies hardness for the combined parameter vertex integrity¹ plus feedback vertex number). On the other hand, notice that MAXNDP is trivially FPT by k ; one can simply reduce it to 2^k instances of NODE-DISJOINT PATHS. A natural question arising from this observation is whether a parameterization by ℓ renders the problem tractable. In this spirit, Marx and Wollan [28] studied this setting and proved that the problem is W[1]-hard even when parameterized by the combined parameter ℓ plus the treewidth of the input graph; a closer look into their proof reveals that their result extends to graphs of bounded pathwidth plus feedback vertex number. This plethora of negative results fails to answer which parameterizations render the problem tractable, and whether a parameterization by ℓ plus some structural parameter (larger than or incomparable to treewidth) may lift it to FPT.

Our Contribution. In the present paper we thoroughly investigate the complexity of MAXNDP under different parameterizations, and determine exactly when it is rendered tractable, by additionally employing the use of approximation in the process (see Figure 1 for a synopsis of our results). We start by showing that the problem is FPT parameterized by the number of vertices of an optimal solution by developing a simple algorithm that makes use of the color-coding technique introduced by Alon, Yuster, and Zwick [2]. We then prove that, albeit simple, this algorithm is in fact sufficient to pinpoint exactly when the problem is fixed-parameter tractable: utilizing a variety of structural observations, we develop FPT algorithms for various parameterizations (most involving ℓ as a parameter) at the core of all of which lies the previously mentioned algorithm. Along the way we also develop an FPT algorithm for the parameterization by the feedback edge number. These positive results, in conjunction with the hardness results of [14, 28], clearly showcase the transition of the problem from FPT to W[1]-hard for its various parameterizations (with the exception of cluster vertex deletion number, where it is unknown whether the problem is W[1]-hard).

Given the apparent hardness of the problem, we move on to consider it under the perspective of parameterized approximation (see [15] for a survey of the area). Here, we observe that utilizing the previously developed FPT algorithms when ℓ is also a parameter, can in fact lead to efficient FPT approximation schemes (FPT-ASes) in case of solely structural

¹ A graph has vertex integrity at most k if there exists a set of at most p vertices such that their deletion results in a graph with connected components of size at most $k - p$.



■ **Figure 1** Our results and hierarchy of the related graph parameters, where vc , vi , td , pw , fvs , cvd , and fes stand for vertex cover, vertex integrity, tree-depth, pathwidth, feedback vertex number, cluster vertex deletion number, and feedback edge number respectively. For any graph, if the parameter at the tail of an arrow is a constant, that is also the case for the one at its head. Green and gray indicate that the problem is FPT (Theorems 2 and 3) and that it admits an FPT approximation scheme (Theorems 4 and 5) respectively. Yellow indicates W[1]-hardness, orange that there is no FPT approximation scheme (Theorem 6), and red XNLP-completeness (Theorem 8). Prior to this work, it was only known that the problem is W[1]-hard parameterized by $vi + fvs$ [14] and by $\ell + pw + fvs$ [28].

parameterizations of the problem, when parameterized by the cluster vertex deletion number, the vertex integrity, or the tree-depth of the input graph; in the latter two cases the problem is known to be W[1]-hard [14].

The FPT approximation schemes developed indicate that the W[1]-hardness can, in some cases, be overcome via the use of approximation. Given the relationship of our studied parameters as well as the FPT-AS for tree-depth, a natural question is whether an analogous approximation scheme exists for the parameterization by pathwidth as well. Notice that the W[1]-hardness by ℓ plus the pathwidth pw and the feedback vertex number fvs of the input graph already excludes the existence of an approximation scheme of running time $f(pw, fvs, \varepsilon)n^{O(1)}$, yet one of time $f(pw, fvs, \varepsilon)n^{g(\varepsilon)}$ remains possible. Our next result is to exclude the existence of such a scheme under the *Parameterized Inapproximability Hypothesis* [26], which was recently proved to hold under the ETH [17], and thus precisely determine the parameter border where the problem transitions from “hard but approximable” to “inapproximable”. By slightly modifying our reduction, we subsequently show that the problem is XNLP-complete when parameterized solely by the pathwidth of the input graph, where XNLP is a complexity class that has been recently brought forth by Bodlaender, Groenland, Nederlof, and Swennenhuis [3], and such a result implies W[t]-hardness for all integers $t \geq 1$.

Lastly, we proceed to a more fine-grained examination of the hardness of MAXNDP when parameterized by the tree-depth of the input graph. Standard dynamic programming techniques can be used to obtain an $n^{O(tw)}$ algorithm, while previous work by Ene, Mnich, Pilipczuk, and Risteski [14] implies that the problem cannot be solved in time $n^{o(\sqrt{td})}$ under the ETH for graphs of tree-depth td , thereby leaving hope for an $n^{o(td)}$ algorithm. We revisit said proof, and by employing a recursive structure introduced by Lampis and Vasilakis [23] we bridge this gap and prove that MAXNDP cannot be solved in time $n^{o(td)}$ under the ETH, rendering the $n^{O(tw)}$ algorithm optimal even for this much smaller class of graphs.

Related Work. Even though MAXNDP has been well-studied under the scope of approximation algorithms, the 20 years old algorithm of ratio $\mathcal{O}(\sqrt{n})$ due to Kolliopoulos and Stein [21] remains the state of the art in general graphs. This has been improved in the case of grid [6] and planar graphs [7], resulting in approximation ratios $\tilde{\mathcal{O}}(n^{1/4})$ and $\tilde{\mathcal{O}}(n^{9/19})$ respectively, where standard $\tilde{\mathcal{O}}$ notation is used to hide polylogarithmic terms. For graphs of pathwidth pw, Ene, Mnich, Pilipczuk, and Risteski [14] have presented an algorithm of approximation ratio $\mathcal{O}(\text{pw}^3)$. Regarding inapproximability results, after a series of works, Chuzhoy, Kim, and Nimavat [9, 10] have shown that the problem cannot be approximated in polynomial time (i) within a factor of $2^{\mathcal{O}(\log^{1-\varepsilon} n)}$ for any constant ε , assuming $\text{NP} \not\subseteq \text{DTIME}(n^{\text{polylog} n})$, and (ii) within a factor of $n^{\mathcal{O}(1/(\log \log n)^2)}$, assuming that there exists some constant $\delta > 0$ such that $\text{NP} \not\subseteq \text{DTIME}(2^{n^\delta})$.

The problem has been also studied under a parameterized complexity perspective. As already noted, it is trivially FPT by k by a simple reduction to 2^k instances of NODE-DISJOINT PATHS, which is well-known to be FPT by the number of demands. On the other hand, Marx and Wollan [28] have shown that it becomes W[1]-hard when parameterized by $\ell + \text{tw}$, with a closer look into their proof revealing that their result extends to the parameterization by $\ell + \text{pw} + \text{fvs}$. Regarding structural parameterizations, Ene, Mnich, Pilipczuk, and Risteski [14] have proved that the problem is W[1]-hard when parameterized by the tree-depth of the input graph; in fact, their proof extends to graphs of bounded vertex integrity plus feedback vertex number. Fleszar, Mnich, and Spoerhase [16] have proposed an algorithm of running time $(k + \text{fvs})^{\mathcal{O}(\text{fvs})} n^{\mathcal{O}(1)}$ as an alternative to the $2^{k \text{fvs}} n^{\mathcal{O}(1)}$ algorithm obtained by reducing the instance to NODE-DISJOINT PATHS and then using Scheffler’s algorithm [32].

Organization. In Section 2 we discuss the general preliminaries. Subsequently, in Section 3 we present various tractability results, followed by the inapproximability result for pathwidth in Section 4. Moving on, in Sections 5 and 6 we present the XNLP-completeness and the refined W[1]-hardness of the problem, when parameterized by the pathwidth and the tree-depth of the input graph respectively. Lastly, in Section 7 we present the conclusion as well as some directions for future research. Proofs of statements marked with (★) are omitted and can be found in the full version.

2 Preliminaries

For $x, y \in \mathbb{Z}$, let $[x, y] = \{z \in \mathbb{Z} : x \leq z \leq y\}$, while $[x] = [1, x]$. For a set S , let $\binom{S}{c}$ denote the set of subsets of S of size c for some $c \in \mathbb{N}$, that is $\binom{S}{c} = \{S' \subseteq S : |S'| = c\}$. Throughout the paper we use standard graph notations [12] and assume familiarity with the basic notions of parameterized complexity [11]. All graphs considered are undirected without loops unless explicitly mentioned otherwise. Let $G = (V, E)$ be a graph. The *cluster vertex deletion number* of G , denoted $\text{cvd}(G)$, is the size of the smallest vertex set whose removal results in a cluster graph, i.e., a union of cliques. The *vertex integrity* of G , denoted $\text{vi}(G)$, is the minimum integer k such that there is a vertex set $S \subseteq V$ such that $|S| + \max_{C \in \text{cc}(G-S)} |V(C)| \leq k$, where $\text{cc}(G-S)$ denotes the set of connected components in $G-S$.

Given a graph $G = (V, E)$ and a partition of V into k independent sets V_1, \dots, V_k , each of size n , k -MULTICOLORED CLIQUE asks whether G contains a k -clique, and is well-known to be W[1]-hard and not to admit any $f(k)n^{o(k)}$ algorithm, where f is any computable function, unless the ETH is false [11]. MULTICOLORED DENSEST k -SUBGRAPH asks for the maximum

number of edges that are induced by a subgraph of G that contains exactly one vertex per independent set (color). Let $E^{i,j} \subseteq E$ denote the set of edges $e = \{u, v\}$ where $u \in V_i$ and $v \in V_j$. In that case, let $s_i = |\{j \in [k] : E^{i,j} \neq \emptyset\}|$ and $\sigma = \sum_{i \in [k]} s_i/2$. Assuming that OPT denotes said maximum value, the *Parameterized Inapproximability Hypothesis* (PIH) [26] states that there exists a constant $0 < c < 1$ such that no $f(k)n^{\mathcal{O}(1)}$ algorithm can distinguish between $\text{OPT} = \sigma$ and $\text{OPT} < c \cdot \sigma$. In fact, one can assume without loss of generality that $1 \leq s_i \leq 3$ for all $i = 1, \dots, k$ [26, Lemma 4.4]. This hypothesis was very recently proved to hold under the ETH [17] (see also [18]), and is the analogue of the PCP theorem in the setting of parameterized complexity.

Lastly, we give a formal definition of the problem this work is concerned with.

MAXIMUM NODE-DISJOINT PATHS

Instance: Graph $G = (V, E)$, set of k demand pairs $\mathcal{M} \subseteq \binom{V}{2}$, and integer $\ell \leq k$.
Goal: Determine whether at least ℓ demand pairs can be routed, where to route a pair we need to select a path connecting it, so that all selected paths are vertex-disjoint.

Notice that in the above definition, even though demand pairs may indeed share a terminal, the paths comprising a feasible solution must be vertex-disjoint, and this constraint also applies to their endpoints. Given an instance $\mathcal{I} = (G, \mathcal{M}, \ell)$ of the optimization (or decision) version of MAXNDP and G' a subgraph of G , $\text{OPT}(\mathcal{I}[G'])$ denotes the maximum number of demands of \mathcal{M} that can be routed in G' . We write $\text{OPT}(\mathcal{I})$ as a shorthand for $\text{OPT}(\mathcal{I}[G])$.

3 FPT Algorithms and Approximation Schemes

Here we present various tractability results for MAXNDP. We start by proving in Section 3.1 that the problem is FPT when parameterized by the number of vertices involved in an optimal solution; using this as well as some structural observations, we obtain FPT algorithms when parameterized by vc , $\ell + \text{cvd}$, $\ell + \text{vi}$, and $\ell + \text{td}$. We additionally develop an FPT algorithm for the parameterization by fes . Moving on, in Section 3.2 we obtain FPT approximation schemes for various structural parameterizations of the problem by making use of the previous FPT algorithms; for most of said parameterizations the problem is known to be W[1]-hard.

3.1 Exact Algorithms

We start with the following theorem.

► **Theorem 1.** *Let $\mathcal{I} = (G, \mathcal{M}, \ell)$ be an instance of MAXNDP. Additionally, let τ be such that there exists a family \mathcal{P} of vertex-disjoint paths each routing a demand of \mathcal{M} , where $|\mathcal{P}| = \min\{\ell, \text{OPT}(\mathcal{I})\}$ and $\sum_{P \in \mathcal{P}} |V(P)| \leq \tau$, with $|V(P)|$ denoting the number of vertices in path P . There is an algorithm that, given \mathcal{I} and τ , decides \mathcal{I} in time $2^{\mathcal{O}(\tau)} n^{\mathcal{O}(1)}$.*

Proof. Let $C = [\tau]$ be a set of τ colors. Randomly color the vertices of G with colors from C , and let A be the event where every one of the at most τ vertices of \mathcal{P} receives a distinct color. Then, it follows that

$$\Pr[A] \geq \frac{\tau!}{\tau^\tau} > \frac{\left(\frac{\tau}{e}\right)^\tau}{\tau^\tau} = e^{-\tau},$$

therefore event A holds with probability at least $e^{-\tau}$.

Now, let $T[S]$ be equal to the maximum number of demands that can be routed by paths using only vertices of colors belonging to $S \subseteq C$, where each color is used in at most one path. Moreover, let $f(S) = 1$ if there exists at least one demand that can be routed using only vertices of colors belonging to S and 0 otherwise. Notice that f can be computed in polynomial time. Then, it holds that

$$T[S] = \max_{S' \subseteq S} \{f(S') + T[S \setminus S']\},$$

thus one can compute $T[C]$ in time $2^{\mathcal{O}(\tau)} n^{\mathcal{O}(1)}$ for a given coloring of the vertices of G .

By repeating this procedure $2^{\mathcal{O}(\tau)}$ times, with high probability there exists some iteration where event A holds, and the total running time is $2^{\mathcal{O}(\tau)} n^{\mathcal{O}(1)}$. By using standard techniques [11, Section 5.6], one can derandomize the described algorithm and obtain a deterministic one of the same running time. \blacktriangleleft

Using Theorem 1 we can obtain various parameterized algorithms, by bounding the number of vertices of an optimal solution using some simple observations.

► **Theorem 2 (\star).** *Given an instance $\mathcal{I} = (G, \mathcal{M}, \ell)$ of MAXNDP, there exist algorithms that decide \mathcal{I} in time*

- $2^{\mathcal{O}(\text{vc})} n^{\mathcal{O}(1)}$,
- $2^{\mathcal{O}(\text{cvd} + \ell)} n^{\mathcal{O}(1)}$,
- $2^{\mathcal{O}(\text{vi}^2 + \text{vi} \cdot \ell)} n^{\mathcal{O}(1)}$,
- $2^{\mathcal{O}(2^{\text{td}} \cdot \ell)} n^{\mathcal{O}(1)}$,

where vc , cvd , vi , and td denote the vertex cover, cluster vertex deletion number, vertex integrity, and tree-depth of G respectively.

Given the W[1]-hardness of MAXNDP when parameterized by the feedback vertex number implied by previous works [14, 28], we move on to consider the parameterization by the feedback edge number, and show that it renders the problem tractable.

► **Theorem 3 (\star).** *Given an instance $\mathcal{I} = (G, \mathcal{M})$ of MAXNDP, there exists an algorithm that computes $\text{OPT}(\mathcal{I})$ in time $3^{\text{fes}} n^{\mathcal{O}(1)}$, where fes denotes the feedback edge number of G .*

3.2 Approximation Schemes

Using the FPT algorithms of Theorem 2, we develop FPT approximation schemes for MAXNDP when parameterized solely by structural parameters.

► **Theorem 4.** *Given an instance $\mathcal{I} = (G, \mathcal{M})$ of MAXNDP, one can $(1 - \varepsilon)$ -approximate $\text{OPT}(\mathcal{I})$ in time $2^{\mathcal{O}(\text{cvd}/\varepsilon)} n^{\mathcal{O}(1)}$ and $2^{\mathcal{O}(\text{vi}^2/\varepsilon)} n^{\mathcal{O}(1)}$, where cvd and vi denote the cluster vertex deletion number and vertex integrity of G respectively.*

Proof. Let $S \subseteq V(G)$ denote the deletion set, which can be computed in time $2^{\mathcal{O}(\text{cvd})} n^{\mathcal{O}(1)}$ for cvd [33] and $2^{\mathcal{O}(\text{vi} \log \text{vi})} n^{\mathcal{O}(1)}$ for vi [13]. Notice that $\text{OPT}(\mathcal{I}[G - S]) \geq \text{OPT}(\mathcal{I}) - |S|$, since every vertex of S can be used to route at most one demand. Consider the case where $\text{OPT}(\mathcal{I}) - |S| \geq (1 - \varepsilon) \text{OPT}(\mathcal{I}) \iff \text{OPT}(\mathcal{I}) \geq |S|/\varepsilon$. Then, $\text{OPT}(\mathcal{I}[G - S]) \geq (1 - \varepsilon) \text{OPT}(\mathcal{I})$. Alternatively, it holds that $\text{OPT}(\mathcal{I}) < |S|/\varepsilon$ and the algorithm of Theorem 2 can be used, with $\ell = \mathcal{O}(|S|/\varepsilon)$.

It remains to compute $\text{OPT}(\mathcal{I}[G - S])$. In the case of the cluster vertex deletion set, $G - S$ is a collection of cliques. In that case, one can compute $\text{OPT}(\mathcal{I}[G - S])$ in polynomial time by reducing to an instance of MAXIMUM MATCHING, on the same vertex set and on

edge set equal to the pairs of \mathcal{M} where both endpoints belong to the same clique. In the case of vertex integrity, it holds that $\text{OPT}(\mathcal{I}[G - S])$ can be computed in FPT time, since it is comprised of $\mathcal{O}(n)$ instances of size at most vi , each of which is solvable in time $2^{\mathcal{O}(\text{vi})}n^{\mathcal{O}(1)}$ due to Theorem 1. \blacktriangleleft

► **Theorem 5.** *Given an instance $\mathcal{I} = (G, \mathcal{M})$ of MAXNDP, one can $(1 - \varepsilon)$ -approximate $\text{OPT}(\mathcal{I})$ in time $2^{\mathcal{O}(2^{\text{td}}/\varepsilon)}n^{\mathcal{O}(1)}$, where td denotes the tree-depth of G .*

Proof. Since G has tree-depth td , one can compute an *elimination forest* of G in time $2^{\mathcal{O}(\text{td}^2)}n^{\mathcal{O}(1)}$ due to [29, 30]. This is a rooted forest on the same vertex set where every pair of vertices adjacent in G adheres to the ancestor/descendant relation. Assume that G is connected, otherwise solve for each connected component. Let r denote the root of the elimination tree, in which case it follows that every connected component of $G - r$ has tree-depth at most $\text{td} - 1$. Notice that it holds that $\text{OPT}(\mathcal{I}[G - r]) \geq \text{OPT}(\mathcal{I}) - 1$, since r can be used to route at most one demand. Let $\varepsilon' < \varepsilon$ and consider the case where

$$\text{OPT}(\mathcal{I}) - 1 \geq \frac{1 - \varepsilon}{1 - \varepsilon'} \cdot \text{OPT}(\mathcal{I}) \iff \text{OPT}(\mathcal{I}) \geq \frac{1 - \varepsilon'}{\varepsilon - \varepsilon'},$$

thus setting $\varepsilon' = \varepsilon/2$ implies that $\text{OPT}(\mathcal{I}) \geq \frac{2 - \varepsilon}{\varepsilon}$. In this case, it holds that an $(1 - \varepsilon')$ -approximation of $\text{OPT}(\mathcal{I}[G - r])$ is an $(1 - \varepsilon)$ -approximation of $\text{OPT}(\mathcal{I})$. On the other hand, if $\text{OPT}(\mathcal{I}) < \frac{2 - \varepsilon}{\varepsilon} = \mathcal{O}(1/\varepsilon)$, we can use the algorithm of Theorem 2, running in time $2^{\mathcal{O}(2^{\text{td}}/\varepsilon)}n^{\mathcal{O}(1)}$.

Consequently, one can recursively argue about the existence of an approximation scheme, since in the case where a graph has tree-depth equal to 1, the instance is polynomial-time solvable. We proceed with bounding the scheme's running time. Let $T(\text{td}, \varepsilon)$ denote the running time for graph G with tree-depth td and error ε , while n' denotes the number of connected components of $G - r$. Notice that it holds that

$$T(\text{td}, \varepsilon) \leq \max\left\{2^{\mathcal{O}(2^{\text{td}}/\varepsilon)}n^{\mathcal{O}(1)}, n' \cdot T(\text{td} - 1, \varepsilon/2)\right\} \leq 2^{\mathcal{O}(2^{\text{td}}/\varepsilon)}n^{\mathcal{O}(1)} + n' \cdot T(\text{td} - 1, \varepsilon/2),$$

while the number of the nodes of the same height in the recursion tree is at most n , since each node corresponds to a connected component. Consequently, it holds that

$$T(\text{td}, \varepsilon) \leq 2^{\mathcal{O}(2^{\text{td}}/\varepsilon)}n^{\mathcal{O}(1)} + n \cdot \sum_{i=1}^{\text{td}} 2^{\mathcal{O}(2^{\text{td}-i} \cdot \frac{1}{\varepsilon/2^i})}n^{\mathcal{O}(1)} = 2^{\mathcal{O}(2^{\text{td}}/\varepsilon)}n^{\mathcal{O}(1)}. \quad \blacktriangleleft$$

4 Inapproximability

Given the FPT approximation scheme of Theorem 5, a natural question arising is whether such an approximation scheme exists for the parameterization by pathwidth as well. Due to the W[1]-hardness of MAXNDP parameterized by $\ell + \text{pw} + \text{fvs}$ [28], there can be no $(1 - \varepsilon)$ -approximation scheme running in time $f(\text{pw}, \text{fvs}, \varepsilon)n^{\mathcal{O}(1)}$, yet one of running time $f(\text{pw}, \text{fvs}, \varepsilon)n^{g(\varepsilon)}$ might be possible.² In this section we answer this question in the negative and prove that there exists some constant $0 < c' < 1$ such that MAXNDP cannot be approximated within a factor of c' in time $f(\text{pw}, \text{fvs})n^{\mathcal{O}(1)}$, for any function f , unless the Parameterized Inapproximability Hypothesis [26] fails.

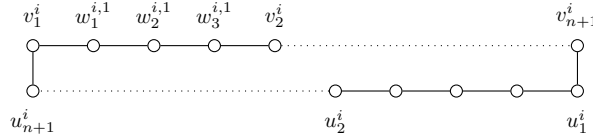
² Assuming there existed such an algorithm of running time $f(\text{pw}, \text{fvs}, \varepsilon)n^{\mathcal{O}(1)}$, setting $\varepsilon < 1/\text{OPT}(\mathcal{I})$ results in obtaining a solution of value at least $(1 - \varepsilon)\text{OPT}(\mathcal{I}) > \text{OPT}(\mathcal{I}) - 1$, i.e., optimal, in time $f(\text{pw}, \text{fvs}, \text{OPT}(\mathcal{I}))n^{\mathcal{O}(1)}$.

► **Theorem 6.** *Assuming the PIH, there exists a constant $c' > 0$ such that MAXNDP does not admit a c' -approximation algorithm of running time $f(\text{pw}, \text{fvs})n^{\mathcal{O}(1)}$.*

Proof. Let $\mathcal{I} = (G, k)$ be an instance of MULTICOLORED DENSEST k -SUBGRAPH, and recall that we assume that G is given to us partitioned into k independent sets V_1, \dots, V_k , where $V_i = \{v_1^i, \dots, v_n^i\}$. Moreover, let $E^{i_1, i_2} \subseteq E(G)$ denote the edges of G with one endpoint in V_{i_1} and the other in V_{i_2} . For every color class $i \in [k]$, let $s_i = |\{j \in [k] : E^{i,j} \neq \emptyset\}|$, and assume without loss of generality that $1 \leq s_i \leq 3$. Set $\sigma = \sum_{i \in [k]} s_i/2$, and notice that $\frac{k}{2} \leq \sigma \leq \frac{3k}{2}$. Let $\text{OPT}(\mathcal{I}) \leq \sigma$ denote the optimal value of instance \mathcal{I} , i.e., the maximum number of edges among the induced subgraphs of G that contain one vertex per color class V_i .

We will construct in polynomial time an instance $\mathcal{J} = (H, \mathcal{M})$ of MAXNDP, where $\text{pw}(H) = \mathcal{O}(k)$, $\text{fvs}(H) = \mathcal{O}(k)$, and $|V(H)| = n^{\mathcal{O}(1)}$, while $\mathcal{M} \subseteq \binom{V(H)}{2}$ is a set of demands. Moreover, it will hold that $\text{OPT}(\mathcal{J}) \leq \ell$, where $\text{OPT}(\mathcal{J})$ denotes the optimal value of instance \mathcal{J} , and $\ell = 2k + \sigma$. We will present a reduction such that (i) if $\text{OPT}(\mathcal{I}) = \sigma$, then $\text{OPT}(\mathcal{J}) = \ell$, and (ii) if $\text{OPT}(\mathcal{I}) < c \cdot \sigma$, then $\text{OPT}(\mathcal{J}) < c' \cdot \ell$, for constants c and c' where $0 < c < 1$ and $c' = \frac{9+c}{10}$.

Choice Gadget. For an independent set V_i , we construct the *choice gadget* \hat{C}_i in the following way. First, for $p \in [n]$, we construct paths on vertex sets $\hat{P}_p^i = \{w_q^{i,p} : q \in [3]\}$, as well as paths \hat{R}_p^i on 3 unnamed vertices each. Then, we introduce vertices v_p^i and u_p^i , for $p \in [n+1]$. Next, we add edges $\{u_{n+1}^i, v_1^i\}$ and $\{v_{n+1}^i, u_1^i\}$. Lastly, for $p \in [n]$, we add edges $\{v_p^i, w_1^{i,p}\}$ and $\{w_3^{i,p}, v_{p+1}^i\}$, as well as an edge from u_p^i to one endpoint of \hat{R}_p^i , and an edge from u_{p+1}^i to the other endpoint of \hat{R}_p^i . See Figure 2 for an illustration. Subsequently, add to \mathcal{M} all pairs (v_p^i, u_{p+1}^i) and (v_p^i, u_{p-1}^i) for $p \in [2, n]$, as well as the pairs (v_1^i, u_2^i) and (v_{n+1}^i, u_n^i) . Let $\mathcal{M}_c \subseteq \mathcal{M}$ denote all such pairs added to \mathcal{M} in this step of the construction. Intuitively, we will consider a one-to-one mapping between the vertex v_p^i of V_i being chosen and the vertices of \hat{P}_p^i not being used to route any of the demands in \mathcal{M}_c .



■ **Figure 2** Choice gadget \hat{C}_i .

Adjacency vertices. Let $E^{i,j} \neq \emptyset$. Introduce an *adjacency vertex* $e_{i,j}$, and add edges $\{e_{i,j}, w_x^{i,p}\}$ and $\{e_{i,j}, w_y^{j,p}\}$ where $p \in [n]$, and $x, y \in [3]$ such that no other adjacency vertex is adjacent to them (since $s_i \leq 3$ for all $i \in [k]$, there always exist such x and y). If $e = \{v_p^i, v_q^j\} \in E^{i,j}$, then add the pair $(w_x^{i,p}, w_y^{j,q})$ in \mathcal{M} . Notice that all adjacency vertices have disjoint neighborhoods, and let $F \subseteq V(H)$ be the set of all adjacency vertices, where $|F| = \sigma$.

This concludes the construction of the instance \mathcal{J} . Notice that $H - F$ is a collection of k choice gadgets, each of which is a cycle. Consequently, the graph obtained from $H - F$ by deleting a vertex per choice gadget is a collection of paths, thus both $\text{pw}(H)$ and $\text{fvs}(H)$ are at most $\mathcal{O}(k)$.

We first prove that if $\text{OPT}(\mathcal{I}) = \sigma$, then $\text{OPT}(\mathcal{J}) = \ell$. Consider a function $h: [k] \rightarrow [n]$ such that $G[\mathcal{V}]$ has σ edges, where $\mathcal{V} = \{v_{h(i)}^i \in V_i : i \in [k]\} \subseteq V(G)$. We construct a family of ℓ vertex-disjoint paths as follows. First, for every $i \in [k]$, we route two demands in \hat{C}_i . In particular, we route the demands $(v_{h(i)}^i, u_{h(i)+1}^i)$ as well as $(v_{h(i)+1}^i, u_{h(i)}^i)$, using

the vertices of the shortest path of \hat{C}_i in each case. Note that in this step we have created $2k$ vertex-disjoint paths connecting terminal pairs belonging to \mathcal{M}_c , and in every gadget \hat{C}_i the only unused vertices are those of $\hat{P}_{h(i)}^i$ and $\hat{R}_{h(i)}^i$. Then, consider the adjacency vertex $e_{i,j}$, where $i, j \in [k]$. Route the demand $(w_x^{i,h(i)}, w_y^{j,h(j)})$ via $e_{i,j}$, since both endpoints of the demand are its neighbors and have not been used in any path so far, for some $x, y \in [3]$. Such a demand indeed exists in \mathcal{M} , since $\{v_{h(i)}^i, v_{h(j)}^j\} \in E^{i,j}$; if that were not the case then $G[\mathcal{V}]$ has less than σ edges. This procedure results in σ additional demands being routed. Notice that since the neighborhoods of all adjacency vertices are disjoint, the ℓ resulting paths are indeed vertex-disjoint.

It remains to prove that if $\text{OPT}(\mathcal{I}) < c \cdot \sigma$, then $\text{OPT}(\mathcal{J}) < c' \cdot \ell$ (or its contrapositive, as we will do in actuality). We start with Claim 7.

▷ **Claim 7.** At most 2 demands can be routed in a choice gadget \hat{C}_i in $H - F$, in which case the only unused vertices are those of \hat{P}_p^i and \hat{R}_p^i , for some $p \in [n]$. Additionally, it holds that $\text{OPT}(\mathcal{J}) \leq \ell$.

Proof. Let $N = 3n + (n + 1)$ and notice that \hat{C}_i is a C_{2N} , thus there are exactly 2 simple paths connecting any pair of its vertices. Moreover, the number of vertices in any simple path between v_p^i and u_p^i , including said endpoints, is $N + 1$. Consequently, to route any demand with both endpoints in \hat{C}_i , either $N + 1 - 4$ or $N + 1 + 4$ vertices are used. In case more than 2 demands are routed, at least $3(N + 1 - 4) = 3N - 9$ vertices are used, which is a contradiction since \hat{C}_i consists of $2N$ vertices.

Assume that exactly 2 demands are routed. Moreover, assume that a demand is routed using $N + 1 + 4$ vertices. Then, both routed demands use at least $(N + 1 + 4) + (N + 1 - 4) > 2N$ vertices, which is a contradiction. Therefore, both demands that are routed in \hat{C}_i use the shortest path connecting their endpoints.

Let (v_p^i, u_q^i) and $(v_{p'}^i, u_{q'}^i)$ denote the demands routed, where $p < p'$ and $p \in [n]$. It holds that $q = p + 1$, since otherwise $q = p - 1$, and $v_{p'}^i$ belongs to the shortest path connecting (v_p^i, u_q^i) , a contradiction. Symmetrically, it follows that $q' = p' - 1$. Lastly, it holds that $q > q'$, since otherwise $u_{q'}^i$ belongs to the shortest path connecting (v_p^i, u_q^i) . Consequently, it follows that $q > q' \iff p + 2 > p'$, which implies that $p < p' < p + 2$, i.e., $p' = p + 1$. In that case, the only unused vertices are those of \hat{P}_p^i and \hat{R}_p^i .

Notice that $H - F$ is a collection of k choice gadgets. Since in each such gadget at most 2 demands can be routed, it follows that $\text{OPT}(\mathcal{J}) \leq 2k + |F| = 2k + \sigma = \ell$. ◁

We now move on to prove that if $\text{OPT}(\mathcal{J}) \geq c' \cdot \ell$, then $\text{OPT}(\mathcal{I}) \geq c \cdot \sigma$. Let \mathcal{P} denote a collection of $\text{OPT}(\mathcal{J})$ vertex-disjoint paths of H routing demands of \mathcal{M} , where $\text{OPT}(\mathcal{J}) \geq c' \cdot \ell$. Additionally, let $\mathcal{C}_j^{\mathcal{P}}$ contain the choice gadgets \hat{C}_i such that there exist *exactly* j paths in \mathcal{P} that route demands in the graph induced by \hat{C}_i , for $j \in [0, 2]$. Notice that $(\mathcal{C}_0^{\mathcal{P}}, \mathcal{C}_1^{\mathcal{P}}, \mathcal{C}_2^{\mathcal{P}})$ defines a partition of the choice gadgets due to Claim 7. Moreover, let $\mathcal{P}^F \subseteq \mathcal{P}$ be comprised of the paths of \mathcal{P} that contain vertices of F . We will say that a path $P \in \mathcal{P}^F$ *intersects* \hat{C}_i if P contains a vertex of \hat{C}_i . We define the *loss* of a solution \mathcal{P} to be $L_{\mathcal{P}} = \ell - |\mathcal{P}|$. Notice that due to Claim 7 it holds that $L_{\mathcal{P}} = 2|\mathcal{C}_0^{\mathcal{P}}| + |\mathcal{C}_1^{\mathcal{P}}| + (|F| - |\mathcal{P}^F|)$.

It holds that $\text{OPT}(\mathcal{J}) = |\mathcal{P}| = \ell - L_{\mathcal{P}} \geq c' \cdot \ell = \ell - (1 - c') \cdot \ell$, thus it follows that $L_{\mathcal{P}} \leq (1 - c') \cdot \ell$. Construct a new solution \mathcal{P}' in the following way: for every $\hat{C}_i \in \mathcal{C}_0^{\mathcal{P}} \cup \mathcal{C}_1^{\mathcal{P}}$, route two demands of \mathcal{M}_c such that there exist two vertex-disjoint paths using only vertices of \hat{C}_i . Afterwards, remove any paths of \mathcal{P}^F that are not vertex-disjoint with those. There are at most 3 vertices of F adjacent to vertices of \hat{C}_i , and by choosing the routed demands of \hat{C}_i in such a way that at least one of those vertices of \hat{C}_i remains unused, it follows that at most 2

paths of \mathcal{P}^F intersect \hat{C}_i . Thus it follows that $L_{\mathcal{P}'} \leq (|F| - |\mathcal{P}^F|) + 2(|\mathcal{C}_0^{\mathcal{P}}| + |\mathcal{C}_1^{\mathcal{P}}|) \leq 2L_{\mathcal{P}}$, therefore $|\mathcal{P}'| = \ell - L_{\mathcal{P}'} \geq \ell - 2L_{\mathcal{P}}$. Furthermore, notice that since for all $i \in [k]$ it holds that $\hat{C}_i \in \mathcal{C}_2^{\mathcal{P}'}$, due to Claim 7 it follows that only the vertices of $\hat{P}_{h(i)}^i$ and $\hat{R}_{h(i)}^i$ remain unused by the paths of \mathcal{P}' that route demands in \mathcal{M}_c , for some function $h: [k] \rightarrow [n]$. Consequently, for any routed demand $(w_x^{i,p}, w_y^{j,q}) \in \mathcal{M} \setminus \mathcal{M}_c$, it holds that $p = h(i)$ and $q = h(j)$.

Let $\mathcal{V} = \{v_{h(i)}^i : i \in [k]\}$, and notice that $|\mathcal{V} \cap V_i| = 1$ for all $i \in [k]$. Let $A = |E(G[\mathcal{V}])|$ denote the number of edges present in the subgraph induced by \mathcal{V} . We will prove that $A \geq c \cdot \sigma$, in which case it follows that $\text{OPT}(\mathcal{I}) \geq c \cdot \sigma$. Notice that $A \geq \ell - 2L_{\mathcal{P}} - 2k = \sigma - 2L_{\mathcal{P}}$, since this is the number of routed demands in $\mathcal{M} \setminus \mathcal{M}_c$ by \mathcal{P}' , while $(w_x^{i,h(i)}, w_y^{j,h(j)}) \in \mathcal{M}$ implies that $\{v_{h(i)}^i, v_{h(j)}^j\} \in E^{i,j}$.

It suffices to prove that $\sigma - 2L_{\mathcal{P}} \geq c \cdot \sigma$. Since $\sigma - 2L_{\mathcal{P}} \geq \sigma - 2(1 - c') \cdot \ell$, we will prove $\sigma - 2(1 - c') \cdot \ell \geq c \cdot \sigma$ instead, which is equivalent to $c' \geq 1 + \frac{\sigma}{2\ell}(c - 1)$. Since $c - 1 < 0$ and $\frac{\sigma}{2\ell} = \frac{1}{2} \cdot \frac{\sigma}{2k + \sigma} \geq \frac{1}{2} \cdot \frac{k/2}{2k + k/2} = \frac{1}{10}$, the statement holds for $c' = 1 + \frac{1}{10}(c - 1) = \frac{9+c}{10}$. ◀

5 XNLP-completeness

The W[1]-hardness results of [14, 28] already imply that MAXNDP is W[1]-hard parameterized by the pathwidth of the input graph. Here we examine in more detail the parameterization solely by pathwidth, and prove that in this case the problem is in fact XNLP-complete. This complexity class was recently brought forth by Bodlaender, Groenland, Nederlof, and Swennenhuis [3] and consists of the parameterized problems such that an instance (x, k) , where x can be encoded with n bits and k denotes the parameter, can be solved non-deterministically in time $f(k)n^{\mathcal{O}(1)}$ and space $f(k) \log n$, for some computable function f .

Such a completeness result in fact implies that MAXNDP parameterized by pathwidth is W[t]-hard for all $t \in \mathbb{N}$. To prove said result, we reduce from the XNLP-complete CHAINED MULTICOLORED CLIQUE, and use a construction quite similar to the one of Theorem 6.

► **Theorem 8 (★).** *MAXNDP parameterized by the pathwidth of the input graph is XNLP-complete.*

6 Refining Hardness for Bounded Tree-depth Graphs

In this section, we refine the hardness result of Ene, Mnich, Pilipczuk, and Risteski [14] for bounded tree-depth graphs, by employing a recursive structure introduced in [23]. The reduction of [14] starts from an instance (G, k) of k -MULTICOLORED CLIQUE, and produces an equivalent instance of MAXNDP on a graph of tree-depth, vertex integrity, and feedback vertex number $\mathcal{O}(k^2)$, implying a $n^{o(\sqrt{\text{td}})}$ lower bound under the ETH. We refine their approach, resulting in a reduction that keeps tree-depth linear in k , thereby improving the lower bound to $n^{o(\text{td})}$ under the ETH. As a consequence of our result, it follows that the standard $n^{\mathcal{O}(\text{tw})}$ algorithm for the problem is optimal, even if one considers the class of graphs of bounded tree-depth.

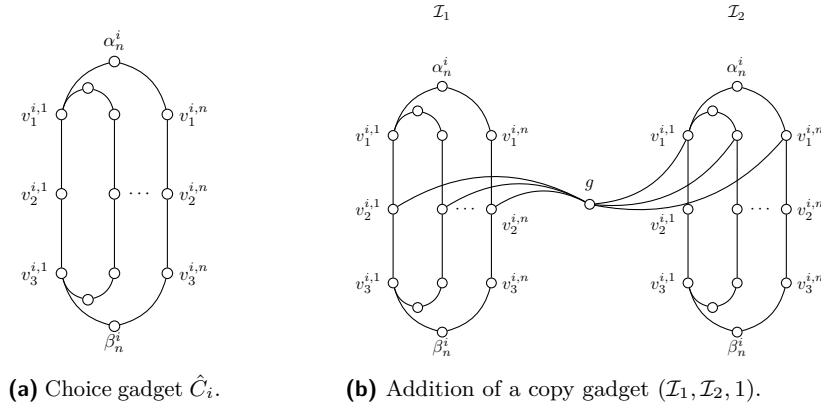
In order to achieve this result, we combine ideas from both [14] and [23]. On a high level, the reduction of [14] consists of k choice gadgets, each of which is used to encode the vertex which is chosen to take part in a supposed clique per color class. Afterwards, it suffices to add $\binom{k}{2}$ vertices in order to verify the existence of edges among all the chosen vertices of the color classes. The deletion of those $\binom{k}{2}$ vertices then gives the bounds for the tree-depth of the graph. In order to avoid this quadratic dependence, we make use of a recursive structure introduced in [23] meant to verify the existence of edges between the chosen vertices, while keeping the tree-depth of the resulting graph linear in k .

► **Theorem 9.** *For any computable function f , if there exists an algorithm that solves MAXNDP in time $f(\text{td})n^{o(\text{td})}$, where td denotes the tree-depth of the input graph, then the ETH is false.*

Proof. Let (G, k) be an instance of k -MULTICOLORED CLIQUE, such that every vertex of G has a self loop, i.e., $\{v, v\} \in E(G)$, for all $v \in V(G)$. Recall that we assume that G is given to us partitioned into k independent sets V_1, \dots, V_k , where $V_i = \{v_1^i, \dots, v_n^i\}$. Assume without loss of generality that $k = 2^z$, for some $z \in \mathbb{N}$ (one can do so by adding dummy independent sets connected to all the other vertices of the graph). Moreover, let $E^{i_1, i_2} \subseteq E(G)$ denote the edges of G with one endpoint in V_{i_1} and the other in V_{i_2} . We will construct in polynomial time an equivalent instance (H, \mathcal{M}, ℓ) of MAXNDP, where H is a graph of tree-depth $\text{td}(H) = \mathcal{O}(k)$, feedback vertex number $\text{fvs}(H) = \mathcal{O}(k^2)$, and size $|V(H)| = n^{\mathcal{O}(1)}$, $\mathcal{M} \subseteq \binom{V(H)}{2}$ is a set of demands, and ℓ is an integer, such that G has a k -clique if and only if at least ℓ demands of \mathcal{M} can be routed in H .

Choice Gadget. For an independent set V_i , we construct the *choice gadget* \hat{C}_i as depicted in Figure 3a. We first construct paths on vertex sets $\hat{P}_j^i = \{v_1^{i,j}, v_2^{i,j}, v_3^{i,j}\}$, where $j \in [n]$. Afterwards, for every $j \in [2, n]$, we introduce vertices α_j^i and β_j^i , connecting them with $v_1^{i,1}, v_1^{i,j}$ and $v_3^{i,1}, v_3^{i,j}$ respectively. Moreover, we add the pair (α_j^i, β_j^i) to \mathcal{M} . Intuitively, we will consider a one-to-one mapping between the vertex v_j^i of V_i belonging to a supposed k -clique of G and the vertices of \hat{P}_j^i not being used to route any of the demands added in this step.

Copy Gadget. Given two instances $\mathcal{I}_1, \mathcal{I}_2$ of a choice gadget \hat{C}_i , when we say that we add a *copy gadget* $(\mathcal{I}_1, \mathcal{I}_2, t)$, where $t \in \{1, 2\}$, we introduce a *copy vertex* g , connect it with all vertices $v_{t+1}^{i,j}$ of \mathcal{I}_1 and all vertices $v_1^{i,j}$ of \mathcal{I}_2 for $j \in [n]$ and add all the pairs $(v_{t+1}^{i,j}, v_1^{i,j})$ to \mathcal{M} .



■ **Figure 3** Choice gadgets and how to copy them.

Adjacency Gadget. Let $i_1, i_2, i'_1, i'_2 \in [k]$. For $i_1 \leq i_2$ and $i'_1 \leq i'_2$, we define the *adjacency gadget*³ $\hat{A}(i_1, i_2, i'_1, i'_2)$ as follows:

- Consider first the case when $i_1 = i_2 = i$ and $i'_1 = i'_2 = i'$. Let the adjacency gadget contain instances of the choice gadgets \hat{C}_i and $\hat{C}_{i'}$, as well as a *validation vertex* $m_{i,i'}$. Add edges between $m_{i,i'}$ and all vertices $v_2^{i,j}$ and $v_2^{i',j}$ for $j \in [n]$. If $e = \{v_j^i, v_{j'}^{i'}\} \in E^{i,i'}$, then add the pair $(v_2^{i,j}, v_2^{i',j'})$ to \mathcal{M} .

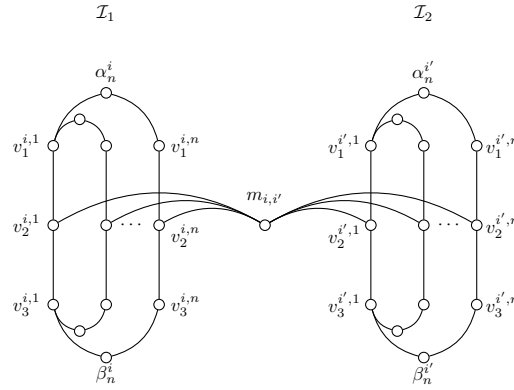
³ For some high-level intuition regarding the adjacency gadgets, we refer the reader to [23, Section 4].

3:12 Parameterized Maximum Node-Disjoint Paths

- Now consider the case when $i_1 < i_2$ and $i'_1 < i'_2$. Then, let $\hat{A}(i_1, i_2, i'_1, i'_2)$ contain instances of choice gadgets \hat{C}_i and $\hat{C}_{i'}$, where $i \in [i_1, i_2]$ and $i' \in [i'_1, i'_2]$, which we will refer to as the *original choice gadgets* of $\hat{A}(i_1, i_2, i'_1, i'_2)$, as well as the adjacency gadgets

- $\hat{A}\left(i_1, \left\lfloor \frac{i_1+i_2}{2} \right\rfloor, i'_1, \left\lfloor \frac{i'_1+i'_2}{2} \right\rfloor\right),$
- $\hat{A}\left(i_1, \left\lfloor \frac{i_1+i_2}{2} \right\rfloor, \left\lceil \frac{i'_1+i'_2}{2} \right\rceil, i'_2\right),$
- $\hat{A}\left(\left\lceil \frac{i_1+i_2}{2} \right\rceil, i_2, i'_1, \left\lfloor \frac{i'_1+i'_2}{2} \right\rfloor\right),$
- $\hat{A}\left(\left\lceil \frac{i_1+i_2}{2} \right\rceil, i_2, \left\lceil \frac{i'_1+i'_2}{2} \right\rceil, i'_2\right).$

Lastly, let \mathcal{I} denote the original choice gadget \hat{C}_p , where $p \in [i_1, i_2] \cup [i'_1, i'_2]$. Notice that there are exactly two instances of choice gadget \hat{C}_p appearing as original choice gadgets in the adjacency gadgets just introduced, say instances \mathcal{I}_1 and \mathcal{I}_2 . Add copy gadgets $(\mathcal{I}, \mathcal{I}_1, 1)$ and $(\mathcal{I}, \mathcal{I}_2, 2)$.



■ **Figure 4** Adjacency gadget $\hat{A}(i, i, i', i')$.

Let graph H be the adjacency gadget $\hat{A}(1, k, 1, k)$ and set $\ell = \gamma \cdot (n - 1) + \delta$, where $\gamma = 2k(2k - 1)$ and $\delta = 5k^2 - 4k$. Notice that it holds that $|V(H)| = (n \cdot k)^{\mathcal{O}(1)}$ as well as that all copy and validation vertices have disjoint neighborhoods. Additionally, let $\mathcal{M}_{\alpha\beta}$ denote the set of pairs of \mathcal{M} of the form (α_j^i, β_j^i) . This concludes the construction of the instance (H, \mathcal{M}, ℓ) .

► **Lemma 10** (\star). *H has the following properties:*

- γ is equal to the number of instances of choice gadgets present in H ,
- δ is equal to the number of copy and validation vertices present in H .

► **Lemma 11** (\star). *It holds that $\text{td}(H) = \mathcal{O}(k)$ and $\text{fvs}(H) = \mathcal{O}(k^2)$.*

► **Lemma 12** (\star). *If G contains a k -clique, then (H, \mathcal{M}, ℓ) is a Yes instance of MAXNDP.*

► **Lemma 13** (\star). *If (H, \mathcal{M}, ℓ) is a Yes instance of MAXNDP, then G contains a k -clique.*

Therefore, in polynomial time, we can construct a graph H , of tree-depth $\text{td} = \mathcal{O}(k)$ and $\text{fvs} = \mathcal{O}(k^2)$ due to Lemma 11, as well as a set of pairs \mathcal{M} , such that, due to Lemmas 12 and 13, deciding whether at least ℓ pairs of \mathcal{M} can be routed is equivalent to deciding whether G has a k -clique. ◀

7 Conclusion

In this work we examine in depth MAXIMUM NODE-DISJOINT PATHS under the perspective of parameterized complexity, painting a complete picture regarding its tractability under most standard parameterizations. We additionally employ the use of approximation and present various (in)approximability results, further enhancing our understanding of the problem in the setting of parameterized approximation.

As a direction for future work, we remark that although Theorem 6 excludes the existence of an approximation scheme, a constant-factor approximation algorithm running in time $f(\text{tw})n^{\mathcal{O}(1)}$ remains possible. Regarding the FPT algorithms of Section 3, all but the one of Theorem 3 rely on providing an upper bound on the number of vertices involved in an optimal solution, leading in some cases to even double-exponential running times. The optimality of those running times is thus a natural open question. Lastly, it is unknown whether the problem is FPT parameterized by the cutwidth or the cluster vertex deletion number of the input graph.

References

- 1 Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Irrelevant vertices for the planar disjoint paths problem. *J. Comb. Theory, Ser. B*, 122:815–843, 2017. doi:10.1016/J.JCTB.2016.10.001.
- 2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
- 3 Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline M. F. Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. *Inf. Comput.*, 300:105195, 2024. doi:10.1016/J.IC.2024.105195.
- 4 Juhi Chaudhary, Harmender Gahlawat, Michal Włodarczyk, and Meirav Zehavi. Kernels for the disjoint paths problem on subclasses of chordal graphs. In *18th International Symposium on Parameterized and Exact Computation, IPEC 2023*, volume 285 of *LIPICs*, pages 10:1–10:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.IPEC.2023.10.
- 5 Kyungjin Cho, Eunjin Oh, and Seunghyeok Oh. Parameterized algorithm for the disjoint path problem on planar graphs: Exponential in k^2 and linear in n . In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, pages 3734–3758. SIAM, 2023. doi:10.1137/1.9781611977554.ch144.
- 6 Julia Chuzhoy and David H. K. Kim. On approximating node-disjoint paths in grids. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015*, volume 40 of *LIPICs*, pages 187–211. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.APPROX-RANDOM.2015.187.
- 7 Julia Chuzhoy, David H. K. Kim, and Shi Li. Improved approximation for node-disjoint paths in planar graphs. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 556–569. ACM, 2016. doi:10.1145/2897518.2897538.
- 8 Julia Chuzhoy, David H. K. Kim, and Rachit Nimavat. Improved approximation for node-disjoint paths in grids with sources on the boundary. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPICs*, pages 38:1–38:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ICALP.2018.38.
- 9 Julia Chuzhoy, David H. K. Kim, and Rachit Nimavat. Almost polynomial hardness of node-disjoint paths in grids. *Theory Comput.*, 17:1–57, 2021. doi:10.4086/toc.2021.v017a006.
- 10 Julia Chuzhoy, David H. K. Kim, and Rachit Nimavat. New hardness results for routing on disjoint paths. *SIAM J. Comput.*, 51(2):17–189, 2022. doi:10.1137/17m1146580.

- 11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 12 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate texts in mathematics*. Springer, 2017. doi:10.1007/978-3-662-53622-3.
- 13 Pål Grønås Drange, Markus S. Dregi, and Pim van 't Hof. On the computational complexity of vertex integrity and component order connectivity. *Algorithmica*, 76(4):1181–1202, 2016. doi:10.1007/S00453-016-0127-X.
- 14 Alina Ene, Matthias Mnich, Marcin Pilipczuk, and Andrej Risteski. On routing disjoint paths in bounded treewidth graphs. In *15th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2016*, volume 53 of *LIPICs*, pages 15:1–15:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.SWAT.2016.15.
- 15 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/A13060146.
- 16 Krzysztof Fleszar, Matthias Mnich, and Joachim Spoerhase. New algorithms for maximum disjoint paths based on tree-likeness. *Math. Program.*, 171(1-2):433–461, 2018. doi:10.1007/s10107-017-1199-3.
- 17 Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Parameterized inapproximability hypothesis under exponential time hypothesis. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024*, pages 24–35. ACM, 2024. doi:10.1145/3618260.3649771.
- 18 Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Almost optimal time lower bound for approximating parameterized clique, csp, and more, under ETH. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025*, pages 2136–2144. ACM, 2025. doi:10.1145/3717823.3718130.
- 19 Richard M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5(4):45–68, 1975. doi:10.1002/NET.1975.5.1.45.
- 20 Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce A. Reed. The disjoint paths problem in quadratic time. *J. Comb. Theory, Ser. B*, 102(2):424–435, 2012. doi:10.1016/j.jctb.2011.07.004.
- 21 Stavros G. Kolliopoulos and Clifford Stein. Approximating disjoint-path problems using packing integer programs. *Math. Program.*, 99(1):63–87, 2004. doi:10.1007/s10107-002-0370-6.
- 22 Tuukka Korhonen, Michal Pilipczuk, and Giannos Stamoulis. Minor containment and disjoint paths in almost-linear time. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024*, pages 53–61. IEEE, 2024. doi:10.1109/FOCS61266.2024.00014.
- 23 Michael Lampis and Manolis Vasilakis. Structural parameterizations for two bounded degree problems revisited. *ACM Trans. Comput. Theory*, 16(3):17:1–17:51, 2024. doi:10.1145/3665156.
- 24 Daniel Lokshantov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018. doi:10.1137/16M1104834.
- 25 Daniel Lokshantov, Pranabendu Misra, Michal Pilipczuk, Saket Saurabh, and Meirav Zehavi. An exponential time parameterized algorithm for planar disjoint paths. *SIAM J. Comput.*, 54(2):321–418, 2025. doi:10.1137/20M1355902.
- 26 Daniel Lokshantov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 2181–2200. SIAM, 2020. doi:10.1137/1.9781611975994.134.
- 27 Daniel Lokshantov, Saket Saurabh, and Meirav Zehavi. Efficient graph minors theory and parameterized algorithms for (planar) disjoint paths. In *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 112–128. Springer, 2020. doi:10.1007/978-3-030-42071-0_9.

- 28 Dániel Marx and Paul Wollan. An exact characterization of tractable demand patterns for maximum disjoint path problems. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*, pages 642–661. SIAM, 2015. doi:10.1137/1.9781611973730.44.
- 29 Wojciech Nadara, Michal Pilipczuk, and Marcin Smulewicz. Computing treedepth in polynomial space and linear FPT time. In *30th Annual European Symposium on Algorithms, ESA 2022*, volume 244 of *LIPICs*, pages 79:1–79:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ESA.2022.79.
- 30 Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. A faster parameterized algorithm for treedepth. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014*, volume 8572 of *Lecture Notes in Computer Science*, pages 931–942. Springer, 2014. doi:10.1007/978-3-662-43948-7_77.
- 31 Neil Robertson and Paul D. Seymour. Graph minors .XIII. The disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006.
- 32 Petra Scheffler. *A practical linear time algorithm for disjoint paths in graphs with bounded tree-width*. TU, Fachbereich 3, 1994.
- 33 Dekel Tsur. Faster parameterized algorithm for cluster vertex deletion. *Theory Comput. Syst.*, 65(2):323–343, 2021. doi:10.1007/S00224-020-10005-W.
- 34 Michal Włodarczyk and Meirav Zehavi. Planar disjoint paths, treewidth, and kernels. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*, pages 649–662. IEEE, 2023. doi:10.1109/FOCS57990.2023.00044.