

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 a 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 \rightarrow Parameterized complexity and exact algorithms

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

Funding This work is partially supported by ANR projects ANR-21-CE48-0022 (S-EX-AP-PE-AL) and ANR-18-CE40-0025-01 (ASSK).

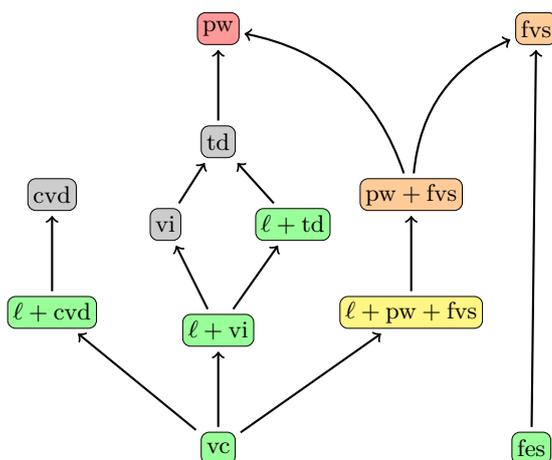
1 Introduction

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, 19, 22, 23, 30, 33] is one of the very first to be proven NP-complete (for k being part of the input) [18], and it has a central role in the field of structural graph theory as well as in parameterized complexity [25], as the breakthrough result by Robertson and Seymour [29] 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, 20], 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 [30] is a key ingredient of the proof of [29]), 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 set 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 [26] 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 set 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

¹ A graph has vertex integrity at most k if the deletion of p vertices results in 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 set number, cluster vertex deletion number, and feedback edge set number respectively. For any graph, if the parameter at the tail of an arrow is a constant then 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 $\ell + pw + fvs$ [26].

all of which lies the previously mentioned algorithm. Along the way we also develop an FPT algorithm for the parameterization by fes . These positive results, in conjunction with the hardness results of [14, 26], 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 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 set 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 $f(pw, \varepsilon)n^{g(\varepsilon)}$ remains possible. Our next result is to exclude the existence of such a scheme under the *Parameterized Inapproximability Hypothesis* [24], 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 $n^{\mathcal{O}(\text{tw})}$ algorithm, while previous work by Ene, Mních, Pilipczuk, and Risteski [14] implies that the problem cannot be solved in time $n^{o(\sqrt{\text{td}})}$ under the ETH for graphs of tree-depth td , thereby leaving hope for an $n^{o(\text{td})}$ algorithm. We revisit said proof, and by employing a recursive structure introduced by Lampis and Vasilakis [21] we bridge this gap and prove that MAXNDP cannot be solved in time $n^{o(\text{td})}$ under the ETH, rendering the $n^{\mathcal{O}(\text{tw})}$ 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 [20] remains the state of the art in general graphs. This has been improved in the case of grid [6] and planar [7] graphs, 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, Mních, 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 [26] 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, Mních, 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 set number. Fleszar, Mních, 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 $2^k \text{fvs}^{\mathcal{O}(\text{fvs})} n^{\mathcal{O}(1)}$ algorithm obtained by reducing the instance to NODE-DISJOINT PATHS and then using Scheffler’s algorithm [30].

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 (\star) are deferred to the appendix.

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 $c \in \mathbb{N}$, i.e. $\binom{S}{c} = \{S' \subseteq S : |S'| = c\}$. Throughout the paper we use standard graph notation [12] and assume familiarity with the basic notions of parameterized complexity [11]. All graphs considered are undirected without loops unless

explicitly mentioned otherwise. For the pathwidth bounds, we use the notion of *mixed search strategy* [31], where an edge is cleared by either placing a searcher on both of its endpoints or sliding one along the edge.

Let G be a graph. A *cluster deletion set* of G is a set $S \subseteq V(G)$ such that every connected component of $G - S$ is a clique, while the *cluster vertex deletion number* of G , denoted by $\text{cvd}(G)$, is the size of its minimum cluster deletion set. A *p-deletion set* of G is a set $S \subseteq V(G)$ such that every connected component of $G - S$ has at most p vertices. The *vertex integrity* of G , denoted by $\text{vi}(G)$, is the minimum k for which there exists a p such that G has a p -deletion set of size $k - p$.

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 -CLIQUE 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, set $s_i = |\{j \in [k] : E^{i,j} \neq \emptyset\}|$, and $\sigma = \sum_{i \in [k]} s_i$. Assuming that OPT denotes said maximum value, the *Parameterized Inapproximability Hypothesis* (PIH) [24] states that there exists a constant $0 < c < 1$ such that no $f(k)n^{O(1)}$ algorithm can distinguish between $\text{OPT} = \sigma$ and $\text{OPT} < c \cdot \sigma$. In fact, as proved in [24], one can assume without loss of generality that $1 \leq s_i \leq 3$ for all $i \in [k]$. This hypothesis was very recently proved to hold under the ETH [17], 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, for most of which it is known to be $W[1]$ -hard, by making use of the previous FPT algorithms.

3.1 Exact Algorithms

We start by presenting the following theorem which will be essential in most of our FPT algorithms.

► **Theorem 1.** *Let $\mathcal{I} = (G, \mathcal{M}, \ell)$ be an instance of MAXNDP. Additionally, let τ 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 $\sum_{P \in \mathcal{P}} |P| \leq \tau$, with $|P|$ 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. ◀

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.** *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.

Proof. We prove the statement by providing bounds on the number of vertices involved in an optimal solution, denoted by τ , and then using the algorithm of Theorem 1. Fix an optimal solution \mathcal{P} , comprised of paths P_i with endpoints s_i and t_i , for $i \in [r]$, where $0 \leq r \leq \ell$ and $(s_i, t_i) \in \mathcal{M}$. We will denote the deletion set in each case by $S \subseteq V(G)$. Note that there is no need to actually compute said deletion set.

In the case of the vertex cover, notice that at least one endpoint of every edge in $P_i \in \mathcal{P}$ belongs to S , while each vertex is involved in at most 2 edges. In that case, it follows that $\tau \leq 3\text{vc}$.

We next consider the case of cluster vertex deletion number. For every path P_i , either both s_i and t_i belong to the same clique of $G - S$ or not. In the first case, there exists an optimal solution that considers the path on vertex set $\{s_i, t_i\}$ instead. In the latter, every such path involves at least one vertex of S , for a total of at most cvd such paths. Moreover,

if such a path involves more than 2 vertices of the same clique of $G - S$, say u_1, u_2, \dots, u_q , indexed by their order of appearance in the path, there exists an optimal solution which is obtained by taking the edge between u_1 and u_q instead. Since every vertex of S in such a path might be neighbors with vertices belonging to at most 2 different cliques of $G - S$, it follows that after “short-cutting” all such paths, at most $\text{cvd} + 4\text{cvd} = 5\text{cvd}$ vertices are used. In total, it follows that $\tau \leq 5\text{cvd} + 2\ell$.

We then consider the case of vertex integrity. For every path P_i , either $P_i \cap S = \emptyset$ or $P_i \cap S \neq \emptyset$. In the first case, it follows that such a path contains at most vi vertices. In the latter, P_i has at least one vertex of S , for a total of at most vi such paths. Moreover, since every vertex of S in P_i might be neighbors with vertices belonging to at most 2 different connected components of $G - S$, at most $\text{vi} + \text{vi}(2\text{vi}) = 2\text{vi}^2 + \text{vi}$ vertices are used. In total, it follows that $\tau \leq 2\text{vi}^2 + \text{vi} + \text{vi} \cdot \ell$.

For tree-depth, it holds that any path in G has length at most 2^{td} , therefore $\tau \leq 2^{\text{td}} \cdot \ell$. ◀

Given the $W[1]$ -hardness of MAXNDP when parameterized by the feedback vertex set number implied by previous works [14, 26], we move on to consider the parameterization by the feedback edge set number, and show that it renders the problem tractable.

► **Theorem 3.** *Given an instance $\mathcal{I} = (G, \mathcal{M})$ of MAXNDP, there exists an algorithm that computes $\text{OPT}(\mathcal{I})$ in time $2^{\mathcal{O}(\text{fes})} n^{\mathcal{O}(1)}$, where fes denotes the feedback edge set number of G .*

Proof. The algorithm will perform branching and reduce the instance to a collection of cycles, in which case the problem is polynomial-time solvable. We first present some reduction rules that are used in their respective order, i.e. Rule 2 is applied after exhaustively applying Rule 1, and Rule 3 after exhaustively applying Rules 1 and 2.

Rule 1. Let $\mathcal{I} = (G, \mathcal{M})$ be an instance of MAXNDP, and $u \in V(G)$ such that $(u, u) \in \mathcal{M}$. Then, replace \mathcal{I} with $\mathcal{I}' = (G - u, \mathcal{M}')$, where $\mathcal{M}' \subseteq \mathcal{M}$ contains the demands of \mathcal{M} whose endpoints both differ from u . It holds that $\text{OPT}(\mathcal{I}) = \text{OPT}(\mathcal{I}') + 1$.

Rule 2. Let $\mathcal{I} = (G, \mathcal{M})$ be an instance of MAXNDP, and $u \in V(G)$ such that $\deg_G(u) = 0$. Then, replace \mathcal{I} with $\mathcal{I}' = (G - u, \mathcal{M}')$, where $\mathcal{M}' \subseteq \mathcal{M}$ contains the demands of \mathcal{M} whose endpoints both differ from u . It holds that $\text{OPT}(\mathcal{I}) = \text{OPT}(\mathcal{I}')$.

Rule 3. Let $\mathcal{I} = (G, \mathcal{M})$ be an instance of MAXNDP, and $u \in V(G)$ such that $\deg_G(u) = 1$, where v denotes its single neighbor, i.e. $N_G(u) = \{v\}$. Then, replace \mathcal{I} with $\mathcal{I}' = (G - u, \mathcal{M}')$, where \mathcal{M}' is obtained by replacing any demand (u, w) in \mathcal{M} with (v, w) . It holds that $\text{OPT}(\mathcal{I}) = \text{OPT}(\mathcal{I}')$.

It is easy to see that applying these rules in their respective order is safe, and let $\mathcal{I} = (G, \mathcal{M})$ denote the instance obtained after exhaustively doing so. Assume without loss of generality that G is connected, otherwise solve each connected component independently. Notice that all vertices of G have degree at least 2. Let $F \subseteq E(G)$ be a feedback edge set of G of minimum size, where $G' = G - F$ is a forest. Since G is a subgraph of the initial graph, it follows that $|F| \leq \text{fes}$.

We will say that $W = \{w_1, \dots, w_r\} \subseteq V(G)$ is an interesting path of G if (i) $G[W]$ induces a path with edges $\{w_i, w_{i+1}\}$ for all $i \in [r - 1]$, (ii) $\deg_G(w_1) \geq 3, \deg_G(w_r) \geq 3$, and (iii) $\deg_G(w_i) = 2$, for all $i \in [2, r - 1]$ (that is, $N_G(w_i) = \{w_{i-1}, w_{i+1}\}$).

We claim that the number of interesting paths of G is $\mathcal{O}(\text{fes})$. To see this, notice that every leaf of G' must be an endpoint of an edge in F , since its degree in G is at least 2. In that case, it follows that the number of leaves in G' is at most $2|F|$. Consequently, the

number of vertices of degree at least 3 in G' is $\mathcal{O}(\text{fes})$, and since G' is a forest, it follows that the number of interesting paths of G' is $\mathcal{O}(\text{fes})$. Since G can be obtained by adding the edges in F to G' , and every such addition increases the number of interesting paths by an additive constant, the statement follows.

Let \mathcal{P} be a maximum-cardinality set of vertex-disjoint paths routing demands of \mathcal{M} in G . Let $v \in V(G)$ such that $\deg_G(v) \geq 3$, where $e_1, e_2, e_3 \in E(G)$ denote three edges adjacent to v . Notice that at least one among those edges, say e_1 , does not take part in \mathcal{P} , that is, $e_1 \not\subseteq P$ for all paths $P \in \mathcal{P}$. Perform branching on all 3 cases and delete the corresponding edge from the graph of the produced instance; by doing so, and potentially applying the reduction rules, the number of the interesting paths is reduced in the 3 produced instances.

Consequently, the branching will cease in at most $\mathcal{O}(\text{fes})$ steps, thus the total number of produced instances is $2^{\mathcal{O}(\text{fes})}$. Let $\mathcal{I}^* = (G^*, \mathcal{M}^*)$ denote one such instance, where every vertex of G^* is of degree exactly 2, i.e. G^* is a collection of cycles. Assume that G^* is connected, otherwise treat its components independently. In that case, it suffices to guess the endpoints of one routed demand and delete the involved vertices (since G^* is a cycle there are only two different such paths), thus reducing G^* into a path where the reduction rules can be applied exhaustively. Therefore, one can compute $\text{OPT}(\mathcal{I}^*)$ in polynomial time. \blacktriangleleft

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 [32] 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 every such clique 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 that 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 $2^{\mathcal{O}(\text{td}^2/\varepsilon)} n^{\mathcal{O}(1)}$, 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 [27, 28]. 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

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

which for $\varepsilon' = \varepsilon/2$ 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

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

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

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

and the statement follows. ◀

4 Inapproximability for Bounded Pathwidth Graphs

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 $\ell + \text{pw} + \text{fvs}$ [26], 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 [24] fails.

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

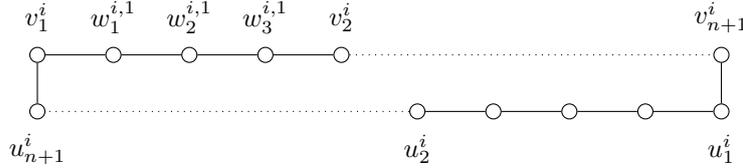
Proof. Let $\mathcal{I} = (G, k)$ be an instance of MULTICOLORED DENSEST k -CLIQUE, and recall that we assume that G is given to us partitioned into k independent sets V_1, \dots, V_k , where $V_i = \{v_1^i, \dots, v_n^i\}$. Moreover, let $E^{i_1, i_2} \subseteq E(G)$ denote the edges of G with one endpoint 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 $\sigma = \sum_{i \in [k]} s_i$, 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

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 $f(\text{pw}, \text{fvs}, \text{OPT}(\mathcal{I})) n^{\mathcal{O}(1)}$.

number of edges among the induced subgraphs of G that contain one vertex x per color class V_i .

We will construct in polynomial time an instance $\mathcal{J} = (H, \mathcal{M})$ of MAXNDP, where $\text{pw}(H) = \mathcal{O}(k^2)$, $\text{fvs}(H) = \mathcal{O}(k^2)$, 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 $c' = \frac{14+c}{15}$.

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 $\mathcal{O}(k^2)$.

We first prove that if $\text{OPT}(\mathcal{I}) = \sigma$, then $\text{OPT}(\mathcal{J}) = \ell$. Consider a function $s : [k] \rightarrow [n]$ such that $G[\mathcal{V}]$ has σ edges, where $\mathcal{V} = \{v_{s(i)}^i \in V_i : i \in [k]\} \subseteq V(G)$. 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 $(v_{s(i)}^i, u_{s(i)+1}^i)$ as well as $(v_{s(i)+1}^i, u_{s(i)}^i)$, 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 $\hat{P}_{s(i)}^i$ and $\hat{R}_{s(i)}^i$. Then, consider the adjacency vertex $e_{i,j}$, where $i, j \in [k]$. Route the demand $(w_x^{i,s(i)}, w_y^{j,s(j)})$ 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 $\{v_{s(i)}^i, v_{s(j)}^j\} \in E^{i,j}$; 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 the following Claim.

▷ **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) , 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 $\mathcal{P}^F \subseteq \mathcal{P}$ contain 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. Since there are at most 3 vertices of F adjacent to vertices of \hat{C}_i , it follows that at most 3 paths of \mathcal{P}^F intersect \hat{C}_i , thus $L_{\mathcal{P}'} \leq (|F| - |\mathcal{P}^F|) + 3(|\mathcal{C}_0^{\mathcal{P}}| + |\mathcal{C}_1^{\mathcal{P}}|) \leq 3L_{\mathcal{P}}$, therefore $|\mathcal{P}'| = \ell - L_{\mathcal{P}'} \geq \ell - 3L_{\mathcal{P}}$. 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 $\hat{P}_{s(i)}^i$ and $\hat{R}_{s(i)}^i$ remain unused by the paths of \mathcal{P}' that route demands in \mathcal{M}_c , for some function $s : [k] \rightarrow [n]$. Consequently, for any routed demand $(w_x^{i,p}, w_y^{j,q}) \in \mathcal{M} \setminus \mathcal{M}_c$, it holds that $p = s(i)$ and $q = s(j)$.

Let $\mathcal{V} = \{v_{s(i)}^i : i \in [k]\}$, 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 $A \geq \ell - 3L_{\mathcal{P}} - 2k = \sigma - 3L_{\mathcal{P}}$, since this is the number of routed demands in $\mathcal{M} \setminus \mathcal{M}_c$ by \mathcal{P}' , while $(w_x^{i,s(i)}, w_y^{j,s(j)}) \in \mathcal{M}$ implies that $\{v_{s(i)}^i, v_{s(j)}^j\} \in E^{i,j}$.

It suffices to prove that $\sigma - 3L_{\mathcal{P}} \geq c \cdot \sigma$. Since $\sigma - 3L_{\mathcal{P}} \geq \sigma - 3(1 - c') \cdot \ell$, the statement holds if $\sigma - 3\ell + 3c'\ell \geq c \cdot \sigma$, which is equivalent to $c' \geq 1 + \frac{\sigma}{3\ell}(c - 1)$. Since $c - 1 < 0$ and $\frac{\sigma}{3\ell} = \frac{1}{3}(1 - \frac{2k}{2k+\sigma}) \leq \frac{1}{3}(1 - \frac{4}{5}) = \frac{1}{15}$, the statement holds for $c' = 1 + \frac{1}{15}(c - 1) = \frac{14+c}{15}$. ◀

5 XNLP-completeness

The W[1]-hardness results of [14, 26] 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.** (\star) 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 [21]. 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 set 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 [21]. 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 [21] 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.** (\star) 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.

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 $f(\text{tw})n^{O(1)}$ 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 cvd .

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. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 193–204. IEEE, 2021. doi:10.1109/FOCS52979.2021.00027.
- 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. New hardness results for routing on disjoint paths. *SIAM J. Comput.*, 51(2):17–189, 2022. doi:10.1137/17m1146580.
- 10 Julia Chuzhoy, David Hong Kyun 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.

- 11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, 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 ETH. *CoRR*, abs/2311.16587, 2023. arXiv:2311.16587, doi:10.48550/ARXIV.2311.16587.
- 18 Richard M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5(4):45–68, 1975. doi:10.1002/NET.1975.5.1.45.
- 19 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.
- 20 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.
- 21 Michael Lampis and Manolis Vasilakis. Structural parameterizations for two bounded degree problems revisited. In *31st Annual European Symposium on Algorithms, ESA 2023*, volume 274 of *LIPICs*, pages 77:1–77:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.ESA.2023.77.
- 22 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018. doi:10.1137/16M1104834.
- 23 Daniel Lokshtanov, Pranabendu Misra, Michal Pilipczuk, Saket Saurabh, and Meirav Zehavi. An exponential time parameterized algorithm for planar disjoint paths. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 1307–1316. ACM, 2020. doi:10.1145/3357713.3384250.
- 24 Daniel Lokshtanov, 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.
- 25 Daniel Lokshtanov, 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.
- 26 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.

- 27 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.
- 28 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.
- 29 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.
- 30 Petra Scheffler. *A practical linear time algorithm for disjoint paths in graphs with bounded tree-width*. TU, Fachbereich 3, 1994.
- 31 Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. Mixed searching and proper-path-width. *Theor. Comput. Sci.*, 137(2):253–268, 1995. doi:10.1016/0304-3975(94)00160-K.
- 32 Dekel Tsur. Faster parameterized algorithm for cluster vertex deletion. *Theory Comput. Syst.*, 65(2):323–343, 2021. doi:10.1007/S00224-020-10005-W.
- 33 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.

A Proofs for Section 5 (XNLP-completeness)

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

Proof. We first argue that MAXNDP parameterized by the pathwidth of the input graph belongs to XNLP. Let $\mathcal{I} = (G, \mathcal{M}, \ell)$ be an instance of MAXNDP, where $n = |\mathcal{I}|$ denotes the number of bits needed to encode \mathcal{I} , and assume that we are also given a path decomposition of G of width pw . We describe a non-deterministic algorithm for \mathcal{I} that uses $\mathcal{O}(\text{pw} \log n)$ bits of memory. Fix an optimal solution, and observe that for each bag, at most pw paths of the solution intersect the bag. The algorithm guesses and stores for each bag both endpoints of the paths of the solution intersecting it ($2\text{pw} \log n$ bits). Moreover, the algorithm guesses and stores for each vertex of the bag an integer from $[0, \ell]$, where 0 indicates that this vertex is not used in the solution, otherwise said vertex is part of the i -th satisfied demand path. For vertices for which we have not stored 0, we also remember their degree, i.e. how many of their neighbors that belong to the same path are present in the bags of the decomposition up to this point (this is an integer in $\{0, 1, 2\}$). In total, $\mathcal{O}(\text{pw} \log n)$ bits are used for this information. It is easy to guess and update these values when introducing a vertex. When Forgetting, the vertex must either have degree 2, or alternatively degree 1 and be an endpoint of its path. If both endpoints of a path have been Forgotten, we remove this from the list of active paths. Keep a counter of how many correct satisfied paths we have seen, using $\mathcal{O}(\log n)$ additional bits, and check in the end that it is at least ℓ .

In order to prove the XNLP-hardness, we present a *parameterized logspace reduction* [3] from CHAINED MULTICOLORED CLIQUE, which is known to be XNLP-complete parameterized by k [3], and is formally defined as follows:

CHAINED MULTICOLORED CLIQUE

- Instance:** Graph $G = (V, E)$, a partition of V into sets V_1, \dots, V_r , as well as a coloring function $f : V \rightarrow [k]$, where for every $\{u, v\} \in E$, if $u \in V_{i_1}$ and $v \in V_{i_2}$, then $|i_1 - i_2| \leq 1$.
- Goal:** Determine whether there exists $W \subseteq V$ such that for every $i \in [r - 1]$, $W \cap (V_i \cup V_{i+1})$ is a clique, and for all $i \in [r]$ and $j \in [k]$, there is a vertex $w \in W \cap V_i$ with $f(w) = j$.

On a high level, CHAINED MULTICOLORED CLIQUE asks whether there exists a clique with $2k$ vertices in $V_i \cup V_{i+1}$ for each $i \in [r - 1]$, containing a vertex of color j both in V_i and in V_{i+1} , for all colors $j \in [k]$. Importantly, the same vertices in V_i are chosen in the clique for both $V_{i-1} \cup V_i$ and $V_i \cup V_{i+1}$. We call such a set a *chained multicolored clique*. Furthermore, one can assume without loss of generality that $|\{v \in V_i : f(v) = j\}| = n$, for every $i \in [r]$ and $j \in [k]$, since one can arbitrarily add a sufficient number of vertices of degree 0.

Let $(G, f, \{V_i : i \in [r]\})$ be an instance of CHAINED MULTICOLORED CLIQUE, and let $V_{i,j} = \{v_1^{i,j}, \dots, v_n^{i,j}\}$ denote the set of vertices belonging to V_i that are of color j . We will construct in polynomial time an equivalent instance (H, \mathcal{M}, ℓ) of MAXNDP, where H is a graph of pathwidth $\text{pw}(H) = \mathcal{O}(k^2)$ and size $|V(H)| = \mathcal{O}(|V(G)| \cdot k)$, $\mathcal{M} \subseteq \binom{V(H)}{2}$ is a set of demands, and $\ell = 2rk + r \binom{k}{2} + (r - 1)k^2$, such that G has a chained multicolored clique if and only if at least ℓ demands of \mathcal{M} can be routed.

The construction will be very similar to the one of Theorem 6, albeit with a few differences. In particular, we will once again employ the use of the choice gadgets introduced there and we refer the reader to Figure 2 for an illustration.

Choice Gadget. For every set $V_{i,j}$, where $i \in [r]$ and $j \in [k]$, we construct the *choice gadget* $\hat{C}_{i,j}$ in the following way. First, for $p \in [n]$, we construct paths on vertex sets $\hat{P}_p^{i,j} = \{w_q^{i,j,p} : q \in [3k]\}$, as well as paths $\hat{R}_p^{i,j}$ on $3k$ unnamed vertices each. Then, we introduce vertices $v_p^{i,j}$ and $u_p^{i,j}$, for $p \in [n+1]$. Next, we add edges $\{u_{n+1}^{i,j}, v_1^{i,j}\}$ and $\{v_{n+1}^{i,j}, u_1^{i,j}\}$. Lastly, for $p \in [n]$, we add edges $\{v_p^{i,j}, w_1^{i,j,p}\}$ and $\{w_{3k}^{i,j,p}, v_{p+1}^{i,j}\}$, as well as an edge from $u_p^{i,j}$ to one endpoint of $\hat{R}_p^{i,j}$, and an edge from $u_{p+1}^{i,j}$ to the other endpoint of $\hat{R}_p^{i,j}$. Subsequently, add to \mathcal{M} all pairs $(v_p^{i,j}, u_{p+1}^{i,j})$ as well as $(v_p^{i,j}, u_{p-1}^{i,j})$, for $p \in [2, n]$, as well as the pairs $(v_1^{i,j}, u_2^{i,j})$ and $(v_{n+1}^{i,j}, u_n^{i,j})$. 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 $v_p^{i,j}$ of $V_{i,j}$ belonging to a supposed chained multicolored clique of G and the vertices of $\hat{P}_p^{i,j}$ not being used to route any of the demands in \mathcal{M}_c .

Adjacency vertices. Next, we introduce some *adjacency vertices* into the graph. For every $i \in [r]$ and $\{j_1, j_2\} \in \binom{[k]}{2}$, introduce vertex e_{j_1, j_2}^i and add edges $\{e_{j_1, j_2}^i, w_{j_2}^{i, j_1, p}\}$ and $\{e_{j_1, j_2}^i, w_{j_1}^{i, j_2, p}\}$ for all $p \in [n]$. Additionally, if $\{v_p^{i, j_1}, v_q^{i, j_2}\} \in E(G)$, then add the pair $(w_{j_2}^{i, j_1, p}, w_{j_1}^{i, j_2, q})$ in \mathcal{M} . Next, for every $i \in [r-1]$ and $j, j' \in [k]$, introduce vertex $e_{j, j'}^{i, i+1}$ and add edges $\{e_{j, j'}^{i, i+1}, w_{2k+j}^{i, j, p}\}$ and $\{e_{j, j'}^{i, i+1}, w_{k+j}^{i+1, j', p}\}$ for all $p \in [n]$. Additionally, if $\{v_p^{i, j}, v_q^{i+1, j'}\} \in E(G)$, then add the pair $(w_{2k+j}^{i, j, p}, w_{k+j}^{i+1, j', q})$ in \mathcal{M} . Notice that all adjacency vertices have disjoint neighborhoods.

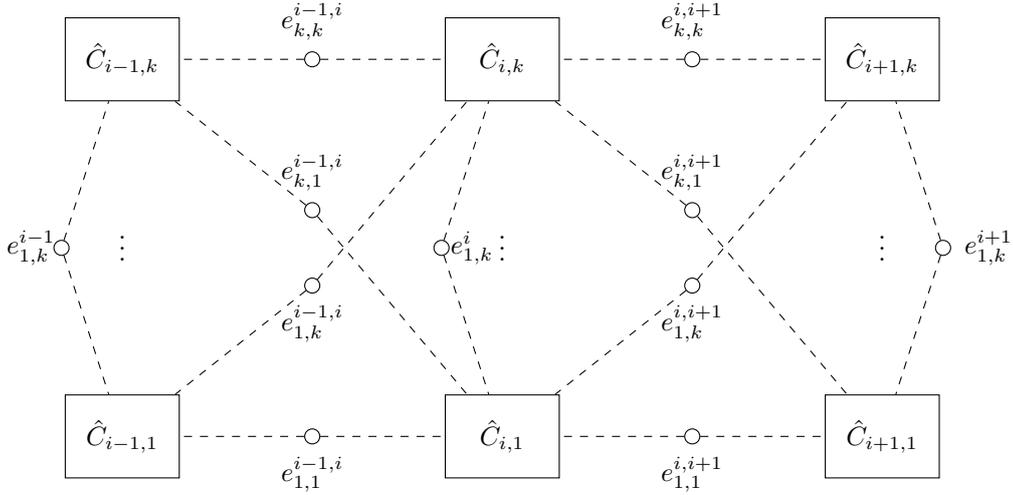
A *set gadget* \hat{S}_i , where $i \in [r]$, is composed of the choice gadgets $\hat{C}_{i,j}$ for all $j \in [k]$ as well as all the adjacency vertices $e_{j, j'}^i$, where $\{j, j'\} \in \binom{[k]}{2}$.

This concludes the construction of H . For an illustration, see Figure 3. It holds that $|V(H)| = rk(6nk + 2n + 2) + r \binom{k}{2} + (r-1)k^2 = \mathcal{O}(rnk^2)$, while $|V(G)| = rnk$, therefore $|V(H)| = \mathcal{O}(|V(G)| \cdot k)$ follows. It remains to argue about the space usage of the algorithm that constructs the instance (H, \mathcal{M}, ℓ) . Notice that in order to uniquely identify the vertices of H , it suffices to keep track of 5 different indices: one identifies the kind of vertex ($u, v, w, e_{j_1, j_2}^i, e_{j, j'}^{i, i+1}$, or a vertex of some path \hat{R} in a choice gadget), and the rest are used to distinguish among vertices of the same kind. In that case, the number of all such encodings is $(nrk)^{\mathcal{O}(1)}$. Moreover, given one such encoding, we can determine the other encodings (i.e. vertices) with which the first either has an edge, or takes part in a demand, since it suffices to only alter some of these indices by some constant, thus the space usage is $\mathcal{O}(\log(nkr))$.

► **Lemma 10.** *It holds that $\text{pw}(H) = \mathcal{O}(k^2)$.*

Proof. For the pathwidth bound, we provide a mixed search strategy to clear H using at most $\mathcal{O}(k^2)$ searchers. Since for the mixed search number ms it holds that $\text{pw}(H) \leq \text{ms}(H)$, it suffices to show that $\text{ms}(H) = \mathcal{O}(k^2)$.

In the following, let j_1, j_2, j , and j' such that $\{j_1, j_2\} \in \binom{[k]}{2}$ and $j, j' \in [k]$. Start with $i = 1$. Place $\binom{k}{2} + k^2$ searchers on vertices e_{j_1, j_2}^1 and $e_{j, j'}^{1, 2}$. Then, use 2 additional searchers to clear first $\hat{C}_{1,1}$, then $\hat{C}_{1,2}$, and so on, until $\hat{C}_{1,k}$ is cleared, and remove the searchers from there. Next, move the searchers from e_{j_1, j_2}^1 to $e_{j_1, j_2}^{1, 2}$, and place k^2 new searchers on the vertices $e_{j, j'}^{1, 2}$. Use the two free searchers to clear all choice gadgets $\hat{C}_{2,1}, \dots, \hat{C}_{2,k}$, and subsequently remove all searchers placed on them as well as on the vertices $e_{j, j'}^{1, 2}$. In order to clear the rest of the graph, we repeat the same procedure. There is no need to introduce any additional searchers, thus $\text{ms}(H) \leq 2 + 2k^2 + \binom{k}{2} = \mathcal{O}(k^2)$. ◀



■ **Figure 3** Part of graph H regarding set gadgets \hat{S}_{i-1} , \hat{S}_i , and \hat{S}_{i+1} . Dashed lines indicate that the respective vertex has neighbors in the corresponding choice gadget.

► **Lemma 11.** *If G contains a chained multicolored clique, then (H, \mathcal{M}, ℓ) is a Yes instance of MAXNDP.*

Proof. Let $\mathcal{W} \subseteq V(G)$ be a chained multicolored clique of G , consisting of vertices $\{v_{s(i,j)}^{i,j} : i \in [r], j \in [k]\}$. We construct a family of ℓ vertex-disjoint paths as follows.

First, for every $i \in [r]$ and $j \in [k]$, we route two demands in $\hat{C}_{i,j}$. In particular, we route the demands $(v_{s(i,j)}^{i,j}, u_{s(i,j)+1}^{i,j})$ as well as $(v_{s(i,j)+1}^{i,j}, u_{s(i,j)}^{i,j})$, using the vertices of the shortest path of $\hat{C}_{i,j}$ in each case. Note that in this step we have created $2rk$ vertex-disjoint paths connecting terminal pairs belonging to \mathcal{M}_c , and in every gadget $\hat{C}_{i,j}$ the only unused vertices are those of $\hat{P}_{s(i,j)}^{i,j}$ and $\hat{R}_{s(i,j)}^{i,j}$.

Then, for $i \in [r]$ and $\{j_1, j_2\} \in \binom{[k]}{2}$, consider the adjacency vertex e_{j_1, j_2}^i . Route the demand $(w_{j_2}^{i, j_1, s(i, j_1)}, w_{j_1}^{i, j_2, s(i, j_2)})$ via said adjacency vertex, since both endpoints of the demand are its neighbors and have not been used in any path so far. This procedure results in $r \binom{k}{2}$ additional demands being routed.

Lastly, for $i \in [r-1]$ and $j, j' \in [k]$, consider the adjacency vertex $e_{j, j'}^{i, i+1}$. Route the demand $(w_{2k+j'}^{i, j, s(i, j)}, w_{k+j}^{i+1, j', s(i+1, j')})$ via said adjacency vertex, since both endpoints of the demand are its neighbors and have not been used in any path so far. Such a demand indeed exists in \mathcal{M} , since $\{v_{s(i,j)}^{i,j}, v_{s(i+1, j')}^{i+1, j'}\} \in E(G)$. This procedure results in $(r-1)k^2$ additional demands being routed.

Notice that since the neighborhoods of all adjacency vertices are disjoint, the ℓ resulting paths are indeed vertex disjoint. ◀

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

Proof. Let $\mathcal{P} = \{P_1, \dots, P_\ell\}$ be a set of ℓ vertex-disjoint paths connecting terminal pairs of \mathcal{M} in H . Assume without loss of generality that said paths are simple, as well as that the only edges among vertices of a path appear between its consecutive vertices. Moreover, define $\mathcal{P}_c = \{P_i \in \mathcal{P} : P_i \text{ routes a demand in } \mathcal{M}_c\}$. Set S to be the set consisting of all the adjacency vertices in H , where $|S| = r \binom{k}{2} + (r-1)k^2$.

Notice that at most $|S|$ paths of \mathcal{P} contain vertices of S , consequently at least $\ell - |S| = 2rk$ paths route demands using only vertices of $H - S$. Moreover, $H - S$ is a collection of rk disconnected choice gadgets. We first prove the following claim.

▷ **Claim 13.** At most 2 demands can be routed in a choice gadget $\hat{C}_{i,j}$ in $H - S$, in which case the only unused vertices are those of $\hat{P}_p^{i,j}$ and $\hat{R}_p^{i,j}$, for some $p \in [n]$.

Proof. Let $N = n(3k) + (n + 1)$ and notice that $\hat{C}_{i,j}$ 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 $v_p^{i,j}$ and $u_p^{i,j}$, including said endpoints, is $N + 1$. Consequently, to route any demand with both endpoints in $\hat{C}_{i,j}$, either $N + 1 - (3k + 1)$ or $N + 1 + (3k + 1)$ vertices are used. In case more than 2 demands are routed, at least $3(N + 1 - 3k - 1)$ vertices are used, which is a contradiction since $\hat{C}_{i,j}$ consists of $2N$ vertices.

Assume that exactly 2 demands are routed. Moreover, assume that a demand is routed using $N + 1 + (3k + 1)$ vertices. Then, both routed demands use at least $N + 1 + (3k + 1) + N + 1 - (3k + 1) > 2N$ vertices, which is a contradiction. Therefore, both demands that are routed in $\hat{C}_{i,j}$ use the shortest path connecting their endpoints.

Let $(v_p^{i,j}, u_q^{i,j})$ and $(v_{p'}^{i,j}, u_{q'}^{i,j})$ denote the demands routed, where $p < p'$ and $p \in [n]$. It holds that $q = p + 1$, since otherwise $q = p - 1$, and $v_{p'}^{i,j}$ belongs to the shortest path connecting $(v_p^{i,j}, u_q^{i,j})$, contradiction. Symmetrically, it follows that $q' = p' - 1$. Lastly, it holds that $q > q'$, since otherwise $u_{q'}^{i,j}$ belongs to the shortest path connecting $(v_p^{i,j}, u_q^{i,j})$. 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 $\hat{P}_p^{i,j}$ and $\hat{R}_p^{i,j}$. ◀

Consequently, due to Claim 13, it follows that exactly $|S|$ paths of \mathcal{P} contain vertices of S . Moreover, in each choice gadget $\hat{C}_{i,j}$, exactly 2 demands are routed, for a total of $2rk$ demands of \mathcal{M}_c , leaving unused only the vertices of $\hat{P}_p^{i,j}$ and $\hat{R}_p^{i,j}$ for some $p \in [n]$. Next we prove that any path not routing a demand of \mathcal{M}_c contains exactly 3 vertices.

▷ **Claim 14.** If $P \in \mathcal{P} \setminus \mathcal{P}_c$, then P is of size 3.

Proof. Since $P \notin \mathcal{P}_c$, it contains a single vertex of S , denoted by s . Let \hat{C}_{i_1, j_1} and \hat{C}_{i_2, j_2} denote the 2 choice gadgets s has neighbors in. Set $S' = S \setminus \{s\}$. Then, it holds that P routes one demand of $\mathcal{M} \setminus \mathcal{M}_c$, using the vertices of the graph $H - S'$. In that case, one endpoint of such a demand can only be a vertex of \hat{C}_{i_1, j_1} and the other of \hat{C}_{i_2, j_2} , and due to the construction of (H, \mathcal{M}, ℓ) , it holds that for every such demand, both its endpoints are neighbors of s . ◀

Let $\mathcal{W} \subseteq V(G)$ be a set of cardinality rk , containing vertex $v_{s(i,j)}^{i,j} \in V_i$ if, for choice gadget $\hat{C}_{i,j}$, it holds that the vertices of $\hat{P}_{s(i,j)}^{i,j}$ are not used to route demands of \mathcal{M}_c . Notice that $|\{w \in W \cap V_i : f(w) = j\}| = 1$, for all $i \in [r]$ and $j \in [k]$. We will prove that \mathcal{W} is a chained multicolored clique of G .

Let i_1, i_2, j_1 and j_2 such that (i) either $i_1 = i_2$ and $\{j_1, j_2\} \in \binom{[k]}{2}$, (ii) or $i_2 = i_1 + 1$ and $j_1, j_2 \in [k]$. Let $v_{s(i_1, j_1)}^{i_1, j_1}, v_{s(i_2, j_2)}^{i_2, j_2}$ belong to \mathcal{W} . Let $P \in \mathcal{P}$ denote the path containing vertex e , where $e = e_{j_1, j_2}^{i_1}$ in case (i) and $e = e_{j_1, j_2}^{i_1, i_2}$ otherwise. Due to Claim 14, it holds that P is comprised of e , as well as two of its neighbors, one in \hat{C}_{i_1, j_1} and one in \hat{C}_{i_2, j_2} . Since the only neighbors of e that are not used by paths in \mathcal{P}_c are $w_{c_1 + j_2}^{i_1, j_1, s(i_1, j_1)}$ and $w_{c_2 + j_1}^{i_2, j_2, s(i_2, j_2)}$, we infer that $(w_{c_1 + j_2}^{i_1, j_1, s(i_1, j_1)}, w_{c_2 + j_1}^{i_2, j_2, s(i_2, j_2)}) \in \mathcal{M}$, where $c_1 = c_2 = 0$ in case (i) and $c_1 = 2k$, $c_2 = k$ otherwise. Consequently, $\{v_{s(i_1, j_1)}^{i_1, j_1}, v_{s(i_2, j_2)}^{i_2, j_2}\} \in E(G)$. Since this holds for any two such vertices belonging to \mathcal{W} , it follows that G has a chained multicolored clique. ◀

Therefore, in polynomial time, we can construct a graph H , where $\text{pw} = \mathcal{O}(k^2)$ due to Lemma 10, as well as a set of pairs \mathcal{M} , such that, due to Lemmas 11 and 12, deciding whether at least ℓ pairs of \mathcal{M} can be routed is equivalent to deciding whether G has a chained multicolored clique. \blacktriangleleft

B Proofs for Section 6 (Refining Hardness for Bounded Tree-depth Graphs)

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

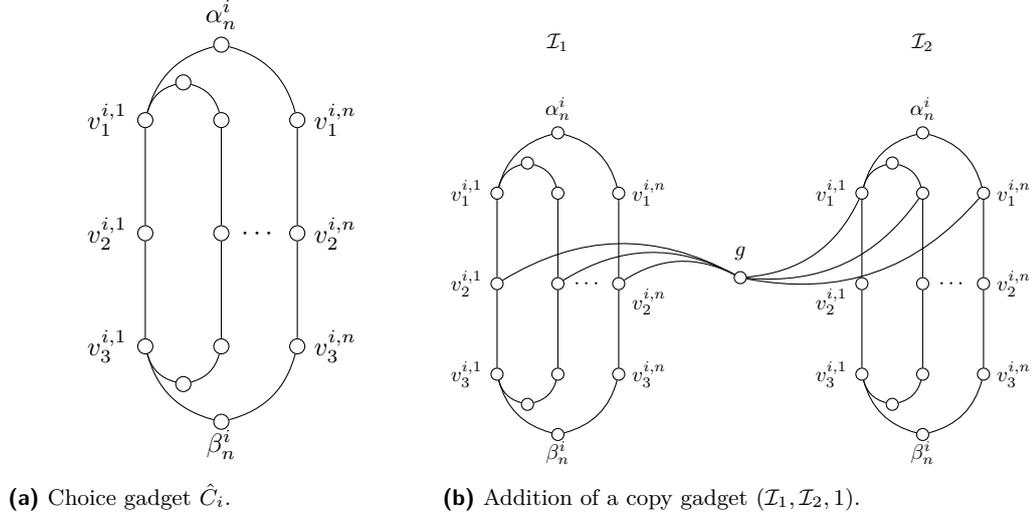
Proof. Let (G, k) be an instance of k -MULTICOLORED CLIQUE, such that every vertex of G has a self loop, i.e. $\{v, v\} \in E(G)$, for all $v \in V(G)$. Recall that we assume that G is given to us partitioned into k independent sets V_1, \dots, V_k , where $V_i = \{v_1^i, \dots, v_n^i\}$. Assume without 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 $\text{td}(H) = \mathcal{O}(k)$, vertex integrity $\text{vi}(H) = \mathcal{O}(k^2)$, feedback vertex set 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 4a. 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 an 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} .

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* $\hat{A}(i_1, i_2, i'_1, i'_2)$ as follows:

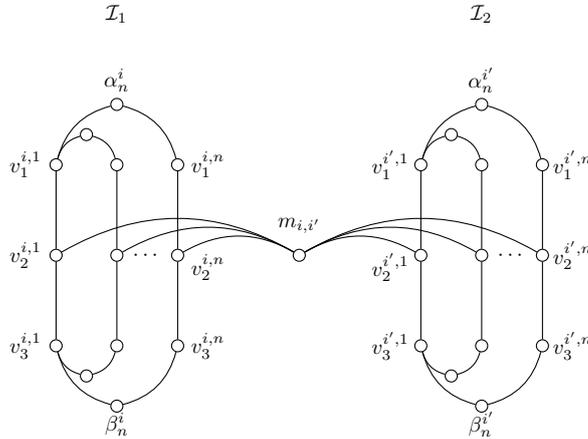
- Consider first the case when $i_1 = i_2 = i$ and $i'_1 = i'_2 = i'$. Let the adjacency gadget contain instances of the choice gadgets \hat{C}_i and $\hat{C}_{i'}$, as well as a *validation vertex* $m_{i,i'}$. Add edges between $m_{i,i'}$ and all vertices $v_2^{i,j}$ and $v_2^{i',j}$ for $j \in [n]$.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



■ **Figure 4** Choice gadgets and how to copy them.

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

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 5** 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 15.** *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.

Proof. First we will prove that for every adjacency gadget $\hat{A}(i_1, i_2, i'_1, i'_2)$ appearing in H, it holds that $i_2 - i_1 = i'_2 - i'_1 = 2^c - 1$, for some $c \in \mathbb{N}$. The statement holds for $\hat{A}(1, k, 1, k)$, as well as when $i_2 - i_1 = i'_2 - i'_1 = 0$. Suppose that it holds for some $\hat{A}(i_1, i_2, i'_1, i'_2)$, i.e. $i_2 - i_1 = i'_2 - i'_1 = 2^c - 1 > 0$, for some $c \in \mathbb{N}$. Then, it follows that $\lfloor \frac{i_1+i_2}{2} \rfloor - i_1 = \lfloor i_2 - 2^{c-1} + 0.5 \rfloor - i_1 = i_2 - i_1 - 2^{c-1} = 2^{c-1} - 1$. Moreover, it follows that $i_2 - \lceil \frac{i_1+i_2}{2} \rceil = i_2 - \lceil i_1 + 2^{c-1} - 0.5 \rceil = i_2 - (i_1 + 2^{c-1}) = 2^{c-1} - 1$. Therefore, the stated property holds.

In that case, for some $\hat{A}(i_1, i_2, i'_1, i'_2)$, in every step of the recursion, intervals $[i_1, i_2]$ and $[i'_1, i'_2]$ are partitioned in the middle, and an adjacency gadget is considered for each of the four combinations. In that case, starting from $\hat{A}(1, k, 1, k)$, there is a single way to produce every adjacency gadget $\hat{A}(i_1, i_1, i_2, i_2)$, where $i_1, i_2 \in [k]$.

For the first statement, notice that the number of instances of choice gadgets is given by the recursive formula $T_1(k) = 2k + 4T_1(k/2)$, where $T_1(1) = 2$. In that case, it follows that

$$T_1(k) = \sum_{i=0}^{\log k} \left(4^i \cdot 2 \cdot \frac{k}{2^i} \right) = 2k \sum_{i=0}^{\log k} 2^i = 2k(2k - 1) = \gamma.$$

For the second statement, notice that the number of copy plus the number of validation vertices is given by the recursive formula $T_2(k) = 4k + 4T_2(k/2)$, where $T_2(1) = 1$. Consequently,

$$\begin{aligned} T_2(k) &= \sum_{i=0}^{\log k-1} \left(4^i \cdot 4 \cdot \frac{k}{2^i} \right) + 4^{\log k} = k^2 + 4k \sum_{i=0}^{\log k-1} 2^i \\ &= k^2 + 4k(k-1) = 5k^2 - 4k = \delta, \end{aligned}$$

and the statement follows. ◀

► **Lemma 16.** *It holds that $\text{td}(H) = \mathcal{O}(k)$, $\text{vi}(H) = \mathcal{O}(k^2)$, and $\text{fvs}(H) = \mathcal{O}(k^2)$.*

Proof. Let $T(\kappa)$ denote the tree-depth of $\hat{A}(i_1, i_2, i'_1, i'_2)$ in the case when $i_2 - i_1 = i'_2 - i'_1 = \kappa$. First, notice that, for $i_1, i_2 \in [k]$, the tree-depth of $\hat{A}(i_1, i_1, i_2, i_2)$ is less than 8. To see this, observe that by removing vertices $m_{i_1, i_2}, v_1^{i_1, 1}, v_3^{i_1, 1}, v_1^{i_2, 1}$ and $v_3^{i_2, 1}$, all remaining connected components are paths of at most 5 vertices. Consequently, $T(1) \leq 8$.

Now, consider the adjacency gadget $\hat{A}(i_1, i_2, i'_1, i'_2)$, where $i_2 - i_1 = i'_2 - i'_1 = \kappa$. This is comprised of adjacency gadgets

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

as well as of exactly 2κ original choice gadgets, each of which is connected with two copy gadgets to other instances of choice gadgets present in the adjacency gadgets. By removing all copy vertices g (see Figure 4b), all the original choice gadgets as well as the adjacency gadgets are disconnected. Therefore, it holds that $T(\kappa) \leq 4\kappa + T(\kappa/2)$, thus, it follows that

$$T(k) \leq 8 + 4 \sum_{i=0}^{\log k-1} \frac{k}{2^i} = \mathcal{O}(k).$$

For the feedback vertex set number and the vertex integrity bounds, let $S \subseteq V(H)$ consist of all the copy and validation vertices. Due to Lemma 15, it holds that $|S| = \delta = \mathcal{O}(k^2)$, while $H - S$ is a collection of γ disconnected choice gadgets. Let $S' = S \cup \{v_1^{i,1}, v_3^{i,1} : \hat{C}_i \text{ in } H\}$, and notice that $H - S'$ is a collection of paths of length at most 5, while $|S'| = \mathcal{O}(k^2)$, thus the statement follows. \blacktriangleleft

► **Lemma 17.** *If G contains a k -clique, then (H, \mathcal{M}, ℓ) is a Yes instance of MAXNDP.*

Proof. Consider $s : [k] \rightarrow [n]$ such that $\mathcal{V} = \{v_{s(i)}^i : i \in [k]\} \subseteq V(G)$ is a k -clique of G . We construct a family of ℓ vertex-disjoint paths as follows.

First, for every instance of \hat{C}_i , where $i \in [k]$, and every $j \in [2, n]$, we route a path from α_j^i to β_j^i through the path \hat{P}_j^i if $j \neq s(i)$; if $j = s(i)$, then we use the path \hat{P}_1^i instead. Note that in this step we have created $\gamma \cdot (n - 1)$ vertex-disjoint paths connecting terminal pairs, and in every gadget \hat{C}_i the only unused vertices are vertices on the path $\hat{P}_{s(i)}^i$.

Then, consider the adjacency gadget $\hat{A}(i, i, i', i')$, where $i, i' \in [k]$. For every such adjacency gadget, we take the 3-vertex path from $v_2^{i, s(i)}$ to $v_2^{i', s(i')}$ through $m_{i, i'}$; note that the assumption that $\{v_{s(i)}^i, v_{s(i')}^{i'}\} \in E(G)$ ensures that $(v_2^{i, s(i)}, v_2^{i', s(i')}) \in \mathcal{M}$.

Lastly, let $(\mathcal{I}_1, \mathcal{I}_2, t)$ be a copy gadget, where $\mathcal{I}_1, \mathcal{I}_2$ are instances of \hat{C}_i and $t \in \{1, 2\}$. For every such copy gadget, we take the 3-vertex path from $v_{t+1}^{i, s(i)}$ of \mathcal{I}_1 to $v_1^{i, s(i)}$ of \mathcal{I}_2 through the copy vertex g connecting them.

Since the neighborhoods of all the copy and validation vertices are disjoint, this is a valid routing, and it follows that exactly $\gamma \cdot (n - 1) + \delta = \ell$ demands are routed. \blacktriangleleft

► **Lemma 18.** *If (H, \mathcal{M}, ℓ) is a Yes instance of MAXNDP, then G contains a k -clique.*

Proof. Let $\mathcal{P} = \{P_1, \dots, P_\ell\}$ be a set of ℓ vertex-disjoint paths connecting terminal pairs of \mathcal{M} in H . Assume without loss of generality that said paths are simple, as well as that the only edges among vertices of the path appear between consecutive vertices. Moreover, let $\mathcal{P}_{\alpha\beta} \subseteq \mathcal{P}$ contain all the paths of \mathcal{P} that route demands of $\mathcal{M}_{\alpha\beta}$. Set S to be the set consisting of all the copy and validation vertices in H , where $|S| = \delta$ due to Lemma 15.

Notice that at most δ paths of \mathcal{P} contain vertices of S , consequently at least $\ell - \delta = \gamma \cdot (n - 1)$ paths route demands using only vertices of $H - S$. Moreover, $H - S$ is a collection of γ disconnected choice gadgets, each of which can be used to route at most $n - 1$ demands. Consequently, it follows that exactly δ paths of \mathcal{P} contain vertices of S , while in each choice gadget, exactly $n - 1$ demands are routed using vertices of $H - S$, thus all the demands of $\mathcal{M}_{\alpha\beta}$ are routed. Let P_i , for $i \in [\delta]$, denote the paths of $\mathcal{P} \setminus \mathcal{P}_{\alpha\beta}$, each of which contains exactly one vertex of S .

Notice that by routing all $n - 1$ demands (α_j^i, β_j^i) in a choice gadget \hat{C}_i , where $j \in [2, n]$, only the vertices of \hat{P}_j^i remain unused, for some $j \in [n]$. To see why this property holds, notice that the shortest path connecting the terminals of such a demand is of length 5, including said terminals. Therefore, at most 3 vertices of a choice gadget may remain unused. However, if said vertices belonged to more than a single path \hat{P}_j^i , then at most $n - 2$ demands can be routed, which is a contradiction. Since every such demand can be routed via either \hat{P}_j^i or \hat{P}_1^i , there exist $n - 1$ paths of $\mathcal{P}_{\alpha\beta}$ that route all demands associated with \hat{C}_i and does not use the vertices of \hat{P}_j^i , for some $j \in [n]$.

Next, we show that every P_i , for $i \in [\delta]$, involves exactly 3 vertices, with one being a vertex of S and the other two being its neighbors. Fix one such i , and let s denote the copy or validation vertex of S appearing in P_i , while \mathcal{I}_1 and \mathcal{I}_2 denote the choice gadgets that have vertices adjacent to s . Set $S' = S \setminus \{s\}$. Then, it holds that P_i routes one demand

of $\mathcal{M} \setminus \mathcal{M}_{\alpha\beta}$, using the vertices of the graph $H - S'$. In that case, one endpoint of such a demand can only be a vertex of \mathcal{I}_1 and the other of \mathcal{I}_2 , and due to the construction of (H, \mathcal{M}, ℓ) , it holds that for every such demand, both its endpoints are neighbors of s .

Next we show that, for $i \in [k]$, for every instance of \hat{C}_i it holds that the only vertices not appearing in paths of $\mathcal{P}_{\alpha\beta}$ are those of $\hat{P}_{s(i)}^{i,j}$, for some function $s : [k] \rightarrow [n]$. Assume there exists a copy gadget $(\mathcal{I}_1, \mathcal{I}_2, t)$, where \mathcal{I}_1 and \mathcal{I}_2 are instances of \hat{C}_i , $t \in \{1, 2\}$ and s is the corresponding copy vertex. Moreover, let $P \in \mathcal{P}$ such that $s \in P$. Then, there exists $s(i) \in [n]$ such that the vertices of $\hat{P}_{s(i)}^i$ are not used to route demands in $\mathcal{M}_{\alpha\beta}$, for both \mathcal{I}_1 and \mathcal{I}_2 , as otherwise no demand of $\mathcal{M} \setminus \mathcal{M}_{\alpha\beta}$ can be routed in the graph induced by \mathcal{I}_1 , \mathcal{I}_2 and s . Furthermore, notice that for every $s_1, s_2 \in S$, $N(s_1) \neq N(s_2)$, while for every instance $\mathcal{I}' \neq \mathcal{I}$ of \hat{C}_i , there is a sequence of copy gadgets $(\mathcal{I}, \mathcal{I}_1, t_1), \dots, (\mathcal{I}_p, \mathcal{I}', t_{p+1})$, where \mathcal{I} denotes the original choice gadget \hat{C}_i in $\hat{A}(1, k, 1, k)$.

Let $\mathcal{V} = \{v_{s(i)}^i : i \in [k]\} \subseteq V(G)$, where $|\mathcal{V} \cap V_i| = 1$, for all $i \in [k]$. We will prove that \mathcal{V} is a clique. Let $v_{s(i_1)}^{i_1}, v_{s(i_2)}^{i_2} \in \mathcal{V}$. Consider the adjacency gadget $\hat{A}(i_1, i_1, i_2, i_2)$ and let $P_i \in \mathcal{P}$ be the path of \mathcal{P} that contains m_{i_1, i_2} . It holds that P_i is comprised of m_{i_1, i_2} , as well as two of its neighbors, one in \mathcal{I}_1 of \hat{C}_{i_1} and one in \mathcal{I}_2 of \hat{C}_{i_2} . Since the only neighbors of m_{i_1, i_2} that are not used by paths in $\mathcal{P}_{\alpha\beta}$ are $v_2^{i_1, s(i_1)}$ and $v_2^{i_2, s(i_2)}$, we infer that $(v_2^{i_1, s(i_1)}, v_2^{i_2, s(i_2)}) \in \mathcal{M}$ and, consequently, $\{v_{s(i_1)}^{i_1}, v_{s(i_2)}^{i_2}\} \in E^{i_1, i_2}$. Since this holds for any two vertices belonging to \mathcal{V} , it follows that G has a k -clique. \blacktriangleleft

Therefore, in polynomial time, we can construct a graph H , of tree-depth $\text{td} = \mathcal{O}(k)$, $\text{vi} = \mathcal{O}(k^2)$ and $\text{fvs} = \mathcal{O}(k^2)$ due to Lemma 16, as well as a set of pairs \mathcal{M} , such that, due to Lemmas 17 and 18, deciding whether at least ℓ pairs of \mathcal{M} can be routed is equivalent to deciding whether G has a k -clique. \blacktriangleleft