SIAM J. DISCRETE MATH.
Vol. 39, No. 1, pp. 92–118

COMPUTING PATHS OF LARGE RANK IN PLANAR FRAMEWORKS DETERMINISTICALLY*

FEDOR V. FOMIN†, PETR A. GOLOVACH†, TUUKKA KORHONEN‡, AND GIANNOS STAMOULIS§

Abstract. A framework consists of an undirected graph G and a matroid M whose elements correspond to the vertices of G. Recently, Fomin et al. [*Proceedings of the* 2023 *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SIAM, 2023, pp. 2214–2227] and Eiben, Koana, and Wahlström [*Proceedings of the* 2024 *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SIAM, 2024, pp. 377–423] developed parameterized algorithms for computing paths of rank k in frameworks. More precisely, for vertices s and t of G, and an integer k, they gave FPT algorithms parameterized by k deciding whether there is an (s,t)-path in G whose vertex set contains a subset of elements of M of rank k. These algorithms are based on the Schwartz–Zippel lemma for polynomial identity testing and thus are randomized, and therefore the existence of a deterministic FPT algorithm for this problem remains open. We present the first deterministic FPT algorithm that solves the problem in frameworks whose underlying graph G is planar. While the running time of our algorithm is worse than the running times of the recent randomized algorithms, our algorithm works on more general classes of matroids. In particular, this is the first FPT algorithm for the case when matroid M is represented over rationals. We complement this result by proving that if the input matroids are given by their independence oracles, then there is no algorithm solving the problem with $f(k)\cdot n^{o(k)}$ oracle queries. Furthermore, this computational lower bound holds even if the input graphs are planar graphs of treewidth at most two.

Key words. planar graph, longest path, linear matroid, irrelevant vertex

MSC codes. 05C38, 05C85, 68R10

DOI. 10.1137/24M1632759

1. Introduction.

A *framework* is a pair (G, M), where G is a graph and $M = (V(G), \mathcal{I})$ is a matroid on the vertex set of G. This term appears in the recent monograph of Lovász [39], where he defines frameworks as graphs with a collection of vectors of $\mathbb{R}^d$ labeling their vertices. Frameworks have appeared in the literature under many different names. For example, they are mentioned as *pregeometric graphs* in the influential work of Lovász [38] on representative families of linear matroids and as *matroid graphs* in the book of Lovász and Plummer [40]. The problem of computing maximum matching in frameworks is closely related to the matchoid, the matroid parity, and polymatroid matching problems (see [40] for an overview). More broadly, the problems of finding specific subgraphs of large ranks in frameworks belong to the wide family of problems about submodular function optimization under combinatorial constraints [8, 9, 43, 15].

*Received by the editors January 19, 2024; accepted for publication (in revised form) September 12, 2024; published electronically January 3, 2025. An extended abstract of this work was published in *Proceedings of ISAAC,* 2023.
https://doi.org/10.1137/24M1632759
Funding: The research leading to these results has received funding from the Research Council of Norway via the project BWCA (grant 314528). Most of the research work for this paper was conducted while the fourth author was affiliated with LIRMM, University of Montpellier, CNRS, Montpellier, France, and supported by the ANR project ESIGMA (ANR-17-CE23-0010) and the French-German Collaboration ANR/DFG Project UTMA (ANR-20-CE92-0027).
†Department of Informatics, University of Bergen, Bergen, 5020 Norway (Fedor.Fomin@uib.no, Petr.Golovach@uib.no).
‡BARC, University of Copenhagen, Copenhagen, Denmark (tuko@di.ku.dk).
§Institute of Informatics, University of Warsaw, Warsaw, Poland (giannos.stamoulis@mimuw.edu.pl).

Fomin et al. in [16] introduced the following Max Rank (s,t)-Path problem. In this problem, given a framework (G,M), two vertices s and t of G, and an integer k, we seek for an (s,t)-path in G where the rank function of M evaluates to at least k. We say that such a path *has rank at least* k.

Max Rank (s,t)-Path

Input:	A framework (G,M), vertices s and t of G, and an integer $k \geq 0$.
Task:	Decide whether G contains an (s,t)-path of rank at least k.

Max Rank (s,t)-Path encompasses several fundamental and well-studied problems about paths and cycles in undirected graphs.

Longest path. Of course, when M is a uniform matroid, then a path is of rank at least k if and only if it contains at least k vertices. In this case, we have the classical Longest Path problem, where for a graph G and integer k the task is to identify whether G contains a path with at least k vertices [2].

T-cycle. In this problem, we are given a set T of terminals and the task is to decide whether there is a cycle through all terminals [6, 25, 48]. T-cycle is the special case of Max Rank (s,t)-Path. Consider the following linear matroid. For every vertex of G not in T we assign a $|T|$-dimensional vector whose all entries are zero. To vertices of T we assign vectors forming an orthonormal basis of $\mathbb{R}^{|T|}$. Then G has a cycle passing through all terminals if and only if (G,M) has an (s,t)-path of rank $|T|$ for some $\{s,t\} \in E(G)$.

Maximum colored path. In the Maximum Colored (s,t)-Path problem, we are given a colored graph G, two vertices s and t of G, and an integer k. The task is to decide whether G has an (s,t)-path containing at least k different colors [7, 16] (see also [10, 11]). Maximum Colored (s,t)-Path is the special case of Max Rank (s,t)-Path where the matroid M is a partition matroid. Indeed, in this matroid the ground set $V(G)$ is partitioned into classes $L_1,\dots,L_t$ and a set I is independent if $|I \cap L_i| \leq 1$ for every label $i \in \{1,\dots,t\}$. In this way, a path of G of rank at least k is a path containing vertices of at least k different (color) classes among $L_1,\dots,L_t$.

1.1. Randomized FPT algorithms for MAX RANK (s,t)-PATH. The parameterized complexity of Max Rank (s,t)-Path was unknown until very recently. The first FPT algorithm for Max Rank (s,t)-Path was given in [16]. This algorithm runs in time $2^{\mathcal{O}(k^2 \log(q+k))} n^{\mathcal{O}(1)}$ and works on frameworks with matroids represented in finite fields of order q. Also, Eiben, Koana, and Wahlström [14], using different techniques, obtained an FPT algorithm for the same problem that runs in time $2^k n^{\mathcal{O}(1)}$ on frameworks with matroids representable over fields of characteristic two. These two algorithms use two different algebraic methods. The algorithm of [16] extends the celebrated algebraic technique based on *cancellation of monomials* used by Björklund, Husfeldt, and Taslaman [6] to solve the T-Cycle problem, while the algorithm of [14] utilizes the toolbox of *(constrained) multilinear detection* [35, 36, 4, 5] combined with *determinantal sieving* [14]. Both these algorithms involve polynomial identity testing and invoke the Schwartz–Zippel lemma, and therefore are randomized. In fact, because of the crucial use of the Schwartz–Zippel lemma in both these algorithms, as the authors of [14] state it, "derandomization appears infeasible" for the algorithms of [16] and [14] for Max Rank (s,t)-Path. Therefore, the next challenge is to obtain *derandomized* FPT algorithms for this problem.

1.2. Our results. Our main result establishes the first *deterministic* FPT algorithm for Max Rank (s,t)-Path on frameworks of planar graphs and matroids representable over finite fields or over the field of rationals.

THEOREM 1.1. *There is a deterministic algorithm that, given a framework* (G, M), *where* G *is a planar graph* G *and* M *is represented as a matrix over a finite field or over* $\mathbb{Q}$, *two vertices* $s, t \in V(G)$ *and an integer* k, *in time* $2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$ *either return an* (s, t)*-path of* G *of rank at least* k *or determine that* G *has no such* (s, t)*-path.*

Note that the randomized FPT algorithms of [16] and [14] work for matroids representable over finite fields or fields of characteristic two. The algorithm of Theorem 1.1, apart from being the first deterministic algorithm for MAX RANK (s, t)-PATH, is also the first FPT algorithm for frameworks whose matroids are not represented over a finite field or a field of characteristic two, but are represented over $\mathbb{Q}$.

We complement this result by demonstrating an unconditional computational lower bound for MAX RANK (s, t)-PATH when the input matroid is given by the independence oracle.

THEOREM 1.2. *There is no algorithm solving* MAX RANK (s, t)-PATH *for frameworks with matroids represented by the independence oracles using* $f(k) \cdot n^{o(k)}$ *oracle calls for any computable function* f. *Furthermore, the lower bound holds for frameworks with planar graphs of treewidth at most two.*

1.3. Our techniques. To design the deterministic FPT algorithm of Theorem 1.1, we follow a different proof strategy than that of [16] and [14]. Our approach is based on the win/win arguments of the celebrated *irrelevant vertex technique* of Robertson and Seymour [45]. The general scheme of this technique is the following. If the graph satisfies certain combinatorial properties, then one can identify a vertex of the graph that can be declared *irrelevant,* meaning that its deletion results in an equivalent instance of the problem. Therefore, after deleting this vertex, we can iterate on the (equivalent) reduced instance. Once this reduction rule cannot be further applied, the obtained reduced instance is equivalent to the original one and is also "simpler." Therefore, it remains to argue that the problem can be solved efficiently in the reduced equivalent instance. This is a standard technique in parameterized algorithms design—see, for example, [17, 24, 28, 42, 33, 23, 29, 20, 25, 27, 30, 26, 19, 3, 47, 22] (see also [12, section 7.8]). The standard measure of complexity of instances for the application of the irrelevant vertex technique is *treewidth.* In particular, the strategy is formulated as follows. As long as the treewidth of the instance is large enough, detect and remove irrelevant vertices. If the treewidth is small, then solve the problem on this equivalent instance using dynamic programming.

Our application of the irrelevant vertex technique is inspired by the algorithm of Kawarabayashi [25] for T-CYCLE and extends its methods. In a typical irrelevant vertex argument, one has to prove that every solution can "avoid" a vertex that will be declared irrelevant. For example, in the classical application of Robertson and Seymour [45] for the DISJOINT PATHS problem, one should argue that (if the graph has large treewidth) any collection of disjoint paths between certain terminals can be "rerouted away" from a vertex v and this vertex should be declared irrelevant. In our case, where we seek an (s, t)-path of *large rank* in a framework, this rerouting should guarantee that large rank is preserved. In general, to deal with such problems on frameworks, one should employ new arguments to adjust this technique to take into account the structure of the matroid. The way we circumvent this problem for MAX RANK (s, t)-PATH is to formulate such a rerouting argument in a "sufficiently insulated" area of the graph where independent sets of the matroid M appear in a homogeneous way. The planarity of the input graph allows one to find such an area using the grid-like structure of *walls.* An overview of this approach is provided in subsection 1.4. This application of the irrelevant vertex technique for frameworks is novel and illustrates an interesting interplay between combinatorial structures and algebraic properties, which may be of independent interest.

The dynamic programming on graphs of bounded treewidth is pretty standard (see, e.g., the book of Cygan et al. [12]) up to one detail. To encode a partial solution, we keep the information about vertices forming independent sets of matroid M visited by a partial solution. However, the number of independent sets of size at most k in M could be of order n^k. Thus a naive encoding of partial solutions would result in blowing-up of the computational complexity. To avoid this, we store only *representative* sets (see [18, 37]) instead of all possible independent sets. Both randomized [18] and deterministic [37] constructions of representative sets require a linear representation of M. This is the reason why Theorem 1.1 is stated for linear matroids. We point out that the dynamic programming subroutine for graphs of bounded treewidth is the only place in the proof of Theorem 1.1 requiring a representation of M.

1.4. Overview of the proof of Theorem 1.1. Our general approach is the following. We show that if the treewidth of the input graph G is $2^{\mathcal{O}(k\log k)}$, then Max Rank (s,t)-Path can be solved in FPT time by a dynamic programming algorithm. Otherwise, if the treewidth is sufficiently large, we give an algorithm that either finds an (s,t)-path of rank at least k or identifies an *irrelevant* vertex v, that is, a vertex whose deletion results in an equivalent instance of the problem. In the latter case, we delete v and iterate on the reduced instance.

If the treewidth of the input graph is large, i.e., of order $2^{\Omega(k\log k)}$, we exploit the grid-minor theorem of Robertson and Seymour for planar graphs [46] that asserts that a planar graph either contains $(w\times w)$-grid as a minor or the treewidth is $\mathcal{O}(w)$. More precisely, we have that given a plane embedding of G, we can find a plane h-wall for $h=2^{\Omega(k\log k)}$ as a topological minor or, equivalently, a plane subgraph of G that is a subdivision of such a wall. To explain our arguments, we need some notions that are informally explained here by making use of figures. In particular, an example of an h-wall for $h=7$ is given in Figure 1.

Note that an h-wall has $\lfloor h/2\rfloor$ nested cycles, called *layers,* that are shown in Figure 1 in red and blue. (Color images are available online.) The layer forming the boundary of a wall is called the *perimeter* of the wall and is shown in red in the figure. We extend the notions of layers and perimeter for a *subdivided* h-wall, that is, the graph obtained from an h-wall by replacing some of its edges by paths. Given a plane subdivided h-wall W in G, we call the subgraph of G induced by the vertices on the perimeter and inside the inner face of the perimeter the *compass* of W and denote it by

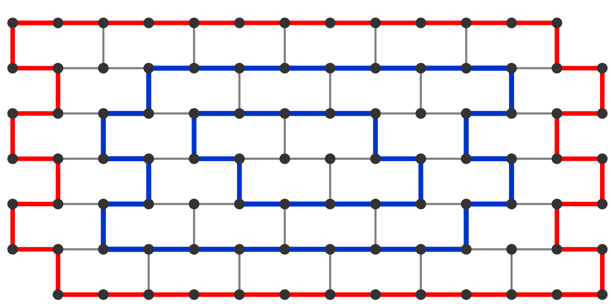

Fig. 1. *A 7-wall and its layers.*

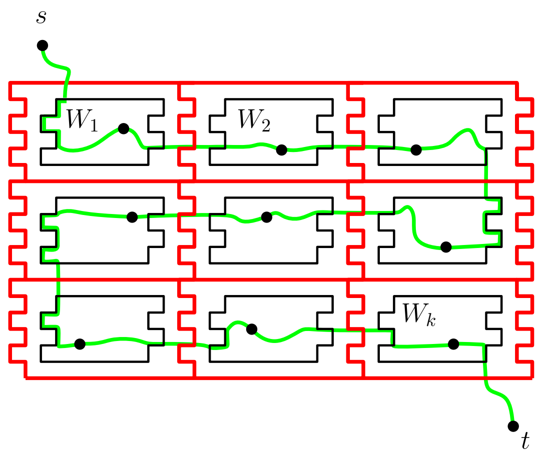

FIG. 2. *An (s,t)-path for walls of big rank.*

$\mathsf{compass}(W)$. Notice that we can assume that the compass of the subdivided h-wall W in G does not contain the terminal vertices s and t by switching to a smaller subwall if necessary. Furthermore, we can assume that $\mathsf{compass}(W)$ is a 2-connected graph as any (s,t)-path can only contain vertices of the biconnected component of $\mathsf{compass}(W)$ containing W. Also we can assume that G has two disjoint paths connecting s and t with two distinct vertices on the perimeter of W; otherwise, any vertex of $\mathsf{compass}(W)$ outside the perimeter is trivially irrelevant.

Observe that for any nontrivial subwall W' of W, $\mathsf{compass}(W')$ is also 2-connected. Therefore, for every two distinct vertices x and y on the perimeter of W' and any $z \in V(\mathsf{compass}(W'))$, $\mathsf{compass}(W')$ has internally disjoint (x,z) and (y,z)-paths. In particular, given a set of vertices $S \subseteq V(\mathsf{compass}(W'))$ that are independent with respect to M, we can join any $z \in S$ with x and y by disjoint paths. This observation is crucial for us.

Suppose that there is a packing of k subwalls $W_1,\dots,W_k$ in W separated by paths in W as shown in Figure 2 such that the rank $r(\mathsf{compass}(W_i)) \geq k$ for $i \in \{1,\dots,k\}$. Then we can choose vertices $v_1,\dots,v_k$ in $\mathsf{compass}(W_1),\dots,\mathsf{compass}(W_k)$, respectively, in such a way that $\{v_1,\dots,v_k\}$ is an independent set of M. Then by our observation, we can construct an (s,t)-path in G that goes through $v_1,\dots,v_k$ as shown in the figure in green. Suppose that this is not the case. Then, by zooming inside the wall, we can assume that $r(\mathsf{compass}(W)) < k$. Moreover, by recursive zooming, we can find a subwall W' of W with the following structural properties (see Figure 3):

- There is a packing of $k+1$ subwalls $W_0,W_1,\dots,W_k$ in W' separated by paths in W' shown in red in Figure 3 such that $r(\mathsf{compass}(W_i)) = r(\mathsf{compass}(W'))$ for $i \in \{1,\dots,k\}$.
- The packing of $W_0,W_1,\dots,W_k$ is surrounded by $\mathcal{O}(k^2)$ "insulation" layers of W' shown in blue.

We claim that vertices of W_0 are irrelevant.

To see this, consider an (s,t)-path P of rank at least k in G. We show that if P goes through a vertex of W_0, then the path can be rerouted as shown in Figure 3 in green to avoid W_0. Consider an independent set $X \subseteq V(P)$ of rank k and let $u_1,\dots,u_\ell$ be the vertices of X that are not spanned by $V(\mathsf{compass}(W'))$ in M. Then $u_1,\dots,u_\ell$ are outside W'. We prove that there are two distinct vertices x and y on the inner insulation layer of W' and an (s,x)-path P_1 and an (y,t)-path P_2 such that (i) x and

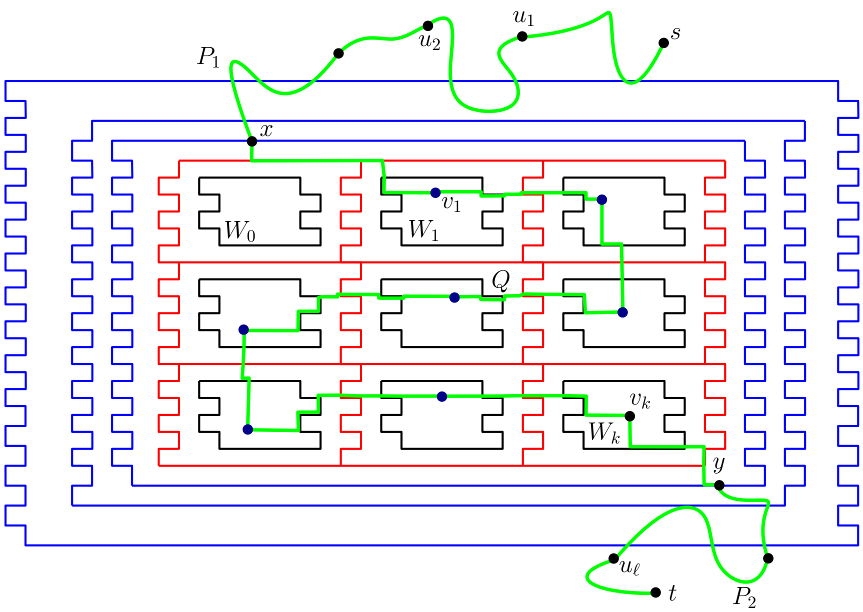

FIG. 3. *Rerouting an* (s,t)*-path.*

y are unique vertices of these paths in the inner insulation layer, and (ii) $u_1,\ldots,u_\ell \in V(P_1)\cup V(P_2)$. The proof that $\mathcal{O}(k^2)$ insulation layers are sufficient for rerouting P is nontrivial. In particular, we adapt the ideas from [25] as well as the structural results of Kleinberg [31]. Further, using the fact that $r(\mathsf{compass}(W_i)) = r(\mathsf{compass}(W'))$ for $i \in \{1,\ldots,k\}$, we show that for every independent set I' of M consisting of vertices in $\mathsf{compass}(W')$, one can also find an independent set I_i of M in $\mathsf{compass}(W_i)$ such that $|I_i| = |I'|$ for every $i \in \{1,\ldots,k\}$. Therefore, one can select, for every W_i, a vertex $v_i \in I_i$ and this choice can be made so that $r(\{v_1,\ldots,v_k\}) = r(\mathsf{compass}(W'))$. Then we construct an (x,y)-path Q in the inner part of W' such that (i) Q is internally disjoint with P_1 and P_2, (ii) Q goes through $v_1,\ldots,v_k$, and (iii) Q avoids W_0. We have that $P' = P_1QP_2$ is an (s,t)-path that goes through $u_1,\ldots,u_\ell$ and $v_1,\ldots,v_k$. Note that, replacing the vertices of X that are spanned by $V(\mathsf{compass}(W'))$ by the vertices $\{v_1,\ldots,v_k\}$, we obtain the set $X' = \{u_1,\ldots,u_\ell,v_1,\ldots,v_k\}$ and $r(X') = r(X)$, and the latter holds since $r(\{v_1,\ldots,v_k\}) = r(\mathsf{compass}(W'))$. Therefore, $r(P') \geq r(X) \geq k$. Since Q avoids W_0, P' has the same property.

Finally, we note that the algorithm of Kawarabayashi [25] for T-CYCLE works for general graphs. The statement of Theorem 1.1 is limited to planar graphs, and planarity is required to ensure that the rerouting does not decrease the rank of an (s,t)-path. It is quite plausible that with additional technicalities our method could be lifted when the underlying graph of the framework is of bounded genus, and more generally, minor-free. However, it is very unclear whether rerouting that does not decrease the rank could be achieved for general graphs. It remains the main obstacle toward pushing the irrelevant vertex technique from frameworks with planar graphs to frameworks with general graphs.

1.5. Organization of the paper. In section 2, we present some basic definitions and preliminary results. In section 3, we show how to reduce to instances of bounded treewidth using the irrelevant vertex technique, while in section 4 we present the dynamic programming algorithm that solves the problem in instances of bounded treewidth. In section 5, we prove the computational lower bound given in Theorem 1.2. We conclude in section 6 with open questions and possible future research directions.

2. Preliminaries. In this section, we introduce basic notation and state some auxiliary results. In subsection 2.1, we provide some basic definitions on parameterized complexity and on graphs, while in subsection 2.2, we give some necessary definitions and results on walls and treewidth. We conclude this section with subsection 2.3, where we provide some useful notions on matroids and the definition of frameworks.

2.1. Basic definitions. We use $\mathbb{Z}_{\geq 1}$ to denote the set of positive integers and $\mathbb{Z}_{\geq 0}$ the set of nonnegative integers. Also, given integers p, q such that $p < q$, we use $[p, q]$ to denote the set $\{p, p+1, \dots, q\}$ and, if $p \geq 1$, we use $[p]$ to denote the set $\{1, \dots, p\}$.

Parameterized complexity. We refer to the book of Cygan et al. [12] for an introduction to the topic. Here we only briefly mention the notions that are most important to state our results. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ^* is a set of strings over a finite alphabet Σ. An input of a parameterized problem is a pair (x, k), where x is a string over Σ and $k \in \mathbb{N}$ is a *parameter.* A parameterized problem is *fixed-parameter tractable* (FPT) if it can be solved in time $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable function f. The complexity class FPT contains all FPT parameterized problems.

Graphs. We use standard graph-theoretic terminology and refer to the textbook of Diestel [13] for missing notions. We consider only finite graphs, and the considered graphs are assumed to be undirected if it is not explicitly said to be otherwise. For a graph G, $V(G)$ and $E(G)$ are used to denote its vertex and edge sets, respectively. Throughout the paper we use $|G| = |V(G)|$. For a graph G and a subset $X \subseteq V(G)$ of vertices, we write $G[X]$ to denote the subgraph of G induced by X. For a vertex v, we denote by $N_G(v)$ the *(open) neighborhood* of v, i.e., the set of vertices that are adjacent to v in G. For $X \subseteq V(G)$, $N_G(X) = \left(\bigcup_{v \in X} N_G(v)\right) \setminus X$. The *degree* of a vertex v is $d_G(v) = |N_G(v)|$.

A walk W of length ℓ in G is a sequence of vertices $v_1, v_2, \dots, v_\ell$, where $v_i v_{i+1} \in E(G)$ for all $1 \leq i < \ell$. The vertices v_1 and v_ℓ are the *endpoints* of W and the vertices $v_2, \dots, v_{\ell-1}$ are the *internal* vertices of W. A path is a walk where no vertex is repeated. For a path P with endpoints s and t, we say that P is an (s,t)-path. A cycle is a path with the additional property that $v_\ell v_1 \in E(G)$ and $\ell \geq 3$.

2.2. Walls and treewidth.

Walls. Let $k, r \in \mathbb{N}$. The $(k \times r)$-*grid* is the graph whose vertex set is $[k] \times [r]$ and two vertices (i, j) and (i', j') are adjacent if and only if $|i - i'| + |j - j'| = 1$. An *elementary r-wall,* for some odd integer $r \geq 3$, is the graph obtained from a $(2r \times r)$-grid with vertices $(x, y) \in [2r] \times [r]$, after the removal of the "vertical" edges $\{(x, y), (x, y+1)\}$ for odd $x + y$, and then the removal of all vertices of degree one. Notice that, as $r \geq 3$, an elementary r-wall is a planar graph that has a unique (up to topological isomorphism) embedding in the plane such that all its finite faces are incident to exactly six edges. The *perimeter* of an elementary r-wall is the cycle bounding its infinite face.

An *r-wall* is any graph W obtained from an elementary r-wall $\bar{W}$ after subdividing edges. We call the vertices that were added after the subdivision operations *subdivision vertices,* while we call the rest of the vertices (i.e., those of $\bar{W}$) *branch vertices.* The *perimeter* of W, denoted by $\mathsf{perim}(W)$, is the cycle of W whose nonsubdivision vertices are the vertices of the perimeter of $\bar{W}$. A *subdivided edge* of W is a path of W whose endpoints are two branch vertices of W and its internal vertices are subdivision vertices of W.

A graph W is a *wall* if it is an r-wall for some odd $r \geq 3$ and we refer to r as the *height* of W. Given a graph G, a *wall of* G is a subgraph of G that is a wall. We insist that, for every r-wall, the number r is always odd. Let W be a wall of a graph G and K' be the connected component of $G \setminus V(\mathsf{perim}(W))$ that contains $W \setminus V(\mathsf{perim}(W))$. We use $\mathsf{inn}(W)$ to denote the graph K'. The *compass* of W, denoted by $\mathsf{compass}(W)$, is the graph $G[V(\mathsf{inn}(W)) \cup V(\mathsf{perim}(W))]$.

The *layers* of an r-wall W, for any odd integer $r \geq 3$, are recursively defined as follows. The first layer of W is its perimeter. For $i = 2, \ldots, (r-1)/2$, the ith layer of W is the $(i-1)$th layer of the wall W' obtained from W after removing from W its perimeter and all occurring vertices of degree one. Notice that each $(2k+1)$-wall has k layers. For every $i = 1, \ldots, (r-1)/2$, we use L_i to denote the ith layer of W. Also, for $i = 2, \ldots, (r-1)/2$ we use $W^{(i)}$ to denote the wall obtained from W after removing from W the layers $L_1, \ldots, L_{i-1}$ and all occurring vertices of degree one and we set $W^{(1)} := W$. Notice that for every $i = 1, \ldots, (r-1)/2$, $\mathsf{perim}(W^{(i)}) = L_i$. See Figure 1 for an example.

Treewidth. A *tree decomposition* of a graph G is a pair $(T, \mathcal{X})$, where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a family of subsets of $V(G)$ such that

- $\bigcup_{t \in V(T)} X_t = V(G)$,
- for every edge e of G there is a $t \in V(T)$ such that X_t contains both endpoints of e, and
- for every $v \in V(G)$, the subgraph of T induced by $\{t \in V(T) \mid v \in X_t\}$ is connected.

The *width* of $(T, \mathcal{X})$ is equal to $\max\{|X_t| - 1 \mid t \in V(T)\}$ and the *treewidth* of G is the minimum width over all tree decompositions of G.

The following result from [21, Lemma 4.2] states that given a $q \in \mathbb{N}$ and a graph G with treewidth more than $9q$, we can find a q-wall of G.

PROPOSITION 2.1. *There exists an algorithm that receives as an input a planar graph G and a $q \in \mathbb{N}$ and outputs, in $2^{q^{\mathcal{O}(1)}} \cdot |G|$ time, either a q-wall W of G or a tree decomposition of G of width at most $9q$.*

2.3. Frameworks. We recall definitions related to frameworks.

Matroids. We refer to the textbook of Oxley [44] for the introduction to matroid theory.

DEFINITION 2.2. *A pair $M = (V, \mathcal{I})$, where V is a* ground set *and $\mathcal{I}$ is a family of subsets of V, called* independent sets of M, *is a* matroid *if it satisfies the following conditions, called* independence axioms*:*

(I1) $\emptyset \in \mathcal{I}$,
(I2) *if $X \subseteq Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$,*
(I3) *if $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then there is $v \in Y \setminus X$ such that $X \cup \{v\} \in \mathcal{I}$.*

An inclusion maximal set of $\mathcal{I}$ is called a *base.* We use $V(M)$ and $\mathcal{I}(M)$ to denote the ground set and the family of independent sets of M, respectively.

Let $M = (V, \mathcal{I})$ be a matroid. We use 2^V to denote the set of all subsets of V. A function $r : 2^V \to \mathbb{Z}_{\geq 0}$ such that for every $X \subseteq V$,

$$r(X) = \max\{|Y| : Y \subseteq X \text{ and } Y \in \mathcal{I}\},$$

is called the *rank function* of M. The *rank of* M, denoted $r(M)$, is $r(V)$; equivalently, the rank of M is the size of any base of M.

Matroid representations. Let $M=(V,\mathcal{I})$ be a matroid and let $\mathbb{F}$ be a field. An $r\times n$-matrix A is a *representation of M over* $\mathbb{F}$ if there is a bijective correspondence f between V and the set of columns of A such that for every $X\subseteq V$, $X\in\mathcal{I}$ if and only if the set of columns $f(X)$ consists of linearly independent vectors of $\mathbb{F}^r$. Equivalently, A is a representation of M if M is isomorphic to the *column* matroid of A, that is, the matroid whose ground set is the set of columns of the matrix and the independence of a set of columns is defined as the linear independence. If M has a such a representation, then M is *representable* over $\mathbb{F}$ and it is also said M is a *linear* (or $\mathbb{F}$-*linear*) matroid. We can assume that the number of rows $r=r(M)$ for a matrix representing M [41].

Whenever we consider a linear matroid, it is assumed that its representation is given and the size of M is $\|M\|=\|A\|$, that is, the bit-length of the representation matrix. Notice that given a representation of a matroid, deciding whether a set is independent demands a polynomial number of field operations. In particular, if the considered field is finite or is the field of rationals, we can verify independence in time that is a polynomial in $\|M\|$. Another standard way to encode a matroid in problem inputs is by using *independence oracles.* Such an oracle, given a subset of the ground set, in unit time correctly returns either yes or no depending on whether the set is independent or not. Thus a matroid can be fully described by its ground set and the independence oracle.

Frameworks. A framework is a pair (G,M), where $M=(V,\mathcal{I})$ is a matroid whose ground set is the set of vertices of G, i.e., $V(M)=V(G)$. An (s,t)-path P in a framework (G,M) has *rank at least* k if there is a set $X\subseteq V(P)$ with $X\in\mathcal{I}$ and $|X|=k$.

3. Rerouting paths and cycles.

In this section, our goal is to prove Theorem 1.1, which we restate here.

THEOREM 1.1. *There is a deterministic algorithm that, given a framework (G,M), where G is a planar graph G and M is represented as a matrix over a finite field or over $\mathbb{Q}$, two vertices $s,t\in V(G)$ and an integer k, in time $2^{2^{\mathcal{O}(k\log k)}}\cdot(|G|+\|M\|)^{\mathcal{O}(1)}$ either returns an (s,t)-path of G of rank at least k or determines that G has no such (s,t)-path.*

The algorithm of Theorem 1.1 consists of two parts. In the first part, we use the irrelevant vertex technique in order to design an algorithm that removes vertices from the input graph as long as its treewidth is big enough. In order to do this, in subsection 3.1 we prove a combinatorial result (Lemma 3.2) that allows us to argue that, given a planar graph and a wall of it and a vertex set S that lies outside the wall, if there is a path P that contains S and invades deeply enough inside the wall, we can find another path P' that contains S (with the same endpoints as P) and avoids some "central area" of the wall. Then, in subsection 3.2, we give an algorithm (Lemma 3.3) that given a planar graph of "big enough" (as a function of k) treewidth, outputs, in time $2^{2^{\mathcal{O}(k\log k)}}\cdot(|G|+\|M\|)^{\mathcal{O}(1)}$, either a path of rank at least k or an irrelevant vertex. Finally, in section 4, we provide the dynamic programming algorithm that solves the problem in graphs of bounded treewidth.

3.1. Rerouting paths and cycles.

In this subsection, we aim to prove the main combinatorial result (Lemma 3.2) that allows us to find an (s,t)-path that contains a given set S and avoids some inner part of a given wall. Before stating Lemma 3.2, we first prove the following result (Lemma 3.1), which will be an important tool for the proof of Lemma 3.2. The proof of Lemma 3.1 is inspired by the proof of [25, Lemma 1].

LEMMA 3.1. *Let G be a planar graph, let $k \in \mathbb{N}$, let W be a wall of height at least $2k+3$, and let $s,t \in V(G)\setminus V(\mathsf{compass}(W))$. Also, let $E = \{e_0, e_1, \ldots, e_k, e_{k+1}\}$ be a set of $k+2$ edges of G with pairwise disjoint endpoints, where, for every $i \in \{1,\ldots,k\}$, $e_i = \{v_i, u_i\}$, $e_0 = \{s', s\}$, $e_{k+1} = \{t', t\}$, and let X be the set $\{s', t'\} \cup \bigcup_{i\in\{1,\ldots,k\}}\{v_i, u_i\}$. If every $v \in X$ is a branch vertex of degree two in W that is a vertex of $\mathsf{perim}(W)$, then there is an (s,t)-path in G that contains the edges $e_0,\ldots,e_{k+1}$ and its intersection with $\mathsf{compass}(W^{(k+1)})$ is a path of $\mathsf{perim}(W^{(k+1)})$ whose endpoints are branch vertices of W.*

Proof. We fix an embedding of G on the plane. Let H be the graph whose vertex set is $\{s,t\} \cup X$ and whose edge set is $\{e_0,\ldots,e_{k+1}\}$. Observe that H is the disjoint union of $k+2$ edges.

We will prove the statement by induction on k. If $k=0$, then $|X| = 2$, and H contains exactly two edges, $e_0 = \{s', s\}$ and $e_1 = \{t', t\}$. By connecting s and t through an (s', t')-path in $\mathsf{perim}(W)$, we obtain the claimed (s,t)-path.

Suppose that $k \geq 1$. Take a vertex $x \in X$, let e_x be the edge of H that is incident to x, and let $x' \in V(H)$ be the other endpoint of e_x. Also, let y be a vertex in $X \setminus \{x, x'\}$ for which there is an (x,y)-path $Q_{x,y}$ in $\mathsf{perim}(W)$ such that no internal vertex of $Q_{x,y}$ is in X. Let $e_y = \{y, y'\}$ be the edge of H that is incident to y. By the choice of y (i.e., $y \in X \setminus \{x, x'\}$), $e_x \neq e_y$. We next perform the following edge contractions: we first contract all edges in $E(Q_{x,y}) \cup \{e_x\}$ to a single vertex. Then, for every $v \in X \setminus \{x, x', y\}$, pick the subdivided edge of W connecting v with the next branch vertex z_v in the perimeter of W (in the clockwise cyclic ordering of the vertices of $\mathsf{perim}(W)$ induced by the cycle $\mathsf{perim}(W)$ in the considered embedding of G). Also, pick the subdivided edge of W connecting z_v with the corresponding vertex w_z on the second layer of W. Then, contract all edges in the two subdivided edges between v, z_v and z_v, w_v. Let G' be the contracted graph. Observe that for every $v \in X \setminus \{x, x', y\}$, w_v is a degree two branch vertex pf $W^{(2)}$ in the perimeter of $W^{(2)}$. Also note that after the contractions, we have a set of $k+1$ edges E' with pairwise disjoint endpoints such that every endpoint that is different from s and t is a degree two branch vertex of $W^{(2)}$ in the perimeter of $W^{(2)}$. Since $W^{(2)}$ has height at least $2k+1$, we can apply the induction hypothesis to the graph G', the wall $W^{(2)}$, and the edge set E' and find an (s,t)-path in G' as claimed. By uncontracting the edges of G', we obtain an (s,t)-path in G with the desired properties. For an illustration of the obtained (s,t)-path, see Figure 4. This completes the proof of the lemma. ∎

We are now ready to prove the following.

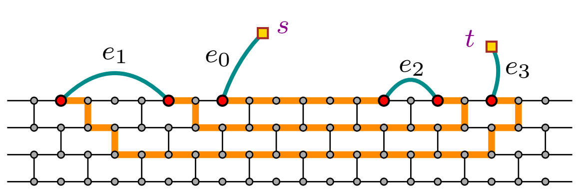

FIG. 4. *Visualization of the proof of Theorem 3.1 for $k=2$. In this example, the edges $e_0,\ldots,e_3$ are depicted in blue and the vertex set X is depicted in red. The highlighted orange paths inside the wall correspond to the paths used in the proof to construct the claimed (s,t)-path.*

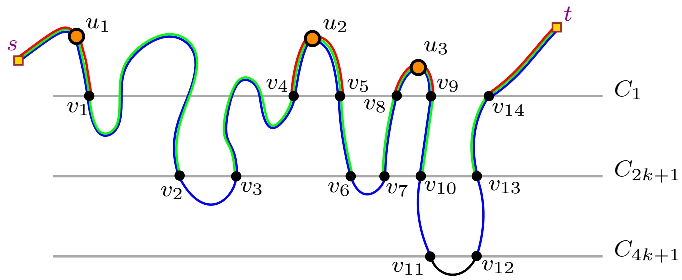

FIG. 5. *An example of an (s,t)-path P containing an independent set $S = \{u_1, u_2, u_3\}$. In this example, $\mathcal{F}_1$ is the collection of the four red paths (the ones with endpoints (s, v_1), (v_4, v_5), (v_8, v_9), and (v_{14}, t)), $\mathcal{F}_2$ is the collection of the four green paths (the ones with endpoints (s, v_2), (v_3, v_6), (v_7, v_{10}), (v_{13}, t)), and $\mathcal{F}_3$ is the collection of the two blue paths (the (s, v_{11})-path and the (v_{12}, t)-path).*

LEMMA 3.2. *There is a function $h : \mathbb{N} \to \mathbb{N}$ such that if $k \in \mathbb{N}$, G is a planar graph, $s, t \in V(G)$, S is a subset of $V(G)$ of size at most k, W is a wall of G of at least $h(k)$ layers and whose compass is disjoint from $S \cup \{s,t\}$, and P is an (s,t)-path of G such that $S \subseteq V(P)$ and P intersects $V(\mathsf{inn}(W^{(h(k))}))$, and then there is an (s,t)-path $\tilde{P}$ of G such that $S \subseteq V(\tilde{P})$ and its intersection with $\mathsf{compass}(W^{(h(k))})$ is a path of $\mathsf{perim}(W^{(h(k))})$ whose endpoints are branch vertices of W. Moreover, $h(k) = \mathcal{O}(k^2)$.*

Proof. We set $h(k) := 2k \cdot (k+2) + 2k + 1$. Let W be a wall of at least $h(k)$ layers. For $i \in [k+2]$, we use C_i to denote the layer $L_{2k\cdot(i-1)+1}$ of W. Intuitively, we take C_1 to be the first layer of W and for every $i \in [2, k+2]$, we take C_i to be the $2k$th consecutive layer after C_{i-1}. Also, we use D_i to denote the vertex set of $\mathsf{compass}(W^{(2k\cdot(i-1)+1)})$. Keep in mind that C_i is the perimeter of $W^{(2k\cdot(i-1)+1)}$. For every $i \in [k+2]$, we consider the collection $\mathcal{F}_i$ of paths of G that are subpaths of P that intersect D_i only on their endpoints and that there is an onto function mapping each vertex $u \in S \cup \{s,t\}$ to the path in $\mathcal{F}_i$ that contains u. Intuitively, for each $u \in S \cup \{s,t\}$ we consider the maximal subpath of P that contains u and intersects D_i only on its endpoints and we define $\mathcal{F}_i$ to be the collection of these maximal paths (see Figure 5 for an example).

Observe that $|\mathcal{F}_1| \leq k+2$ (since $|S \cup \{s,t\}| \leq k+2$) and $|\mathcal{F}_{k+2}| \geq 2$ (since $V(P) \cap V(\mathsf{inn}(W^{(h(k))})) \neq \emptyset$ and therefore P intersects at least twice every C_i, $i \in [k+2]$). For every $i \in [k+2]$, we assume that $\mathcal{F}_i = \{F_{i,1}, \dots, F_{i,|\mathcal{F}_i|}\}$, where the ordering is given by traversing P from s to t. For every $i \in [k+2]$, we set $\mathcal{Q}_i = \{Q_{i,1}, \dots, Q_{i,|\mathcal{F}_i|-1}\}$, where, for each $j \in [|\mathcal{F}_i| - 1]$, $Q_{i,j}$ is the minimal subpath of P that intersects both $V(F_{i,j})$ and $V(F_{i,j+1})$. Observe that, for every $i \in [k+2]$, P is the concatenation of the paths $F_{i,1}, Q_{i,1}, F_{i,2}, \dots, Q_{i,|\mathcal{F}_i|-1}, F_{i,|\mathcal{F}_i|}$. In Figure 5, $\mathcal{Q}_1 = \{Q_{1,1}, Q_{1,2}, Q_{1,3}\}$, where $Q_{1,1}$ is the (v_1, v_4)-subpath, $Q_{1,2}$ is the (v_5, v_8)-subpath, and $Q_{1,3}$ is the (v_9, v_{14})-subpath of P, $\mathcal{Q}_2 = \{Q_{2,1}, Q_{2,2}, Q_{2,3}\}$, where $Q_{2,1}$ is the (v_2, v_3)-subpath, $Q_{2,2}$ is the (v_6, v_7)-subpath, and $Q_{2,3}$ is the (v_{10}, v_{13})-subpath of P, and $\mathcal{Q}_3$ consists of the (v_{11}, v_{12})-subpath $Q_{3,1}$ of P.

It is easy to see that for every $i \in [k+1]$, $|\mathcal{F}_{i+1}|$ is equal to $|\mathcal{F}_i|$ minus the number of paths in $\mathcal{Q}_i$ that do not intersect C_{i+1} and therefore, $|\mathcal{F}_i| \geq |\mathcal{F}_{i+1}|$. Therefore, given that $|\mathcal{F}_1| \leq k+2$, $|\mathcal{F}_{k+2}| \geq 2$, and for every $i \in [k+1]$, $|\mathcal{F}_i| \geq |\mathcal{F}_{i+1}|$, there is an $i_0 \in [k+1]$ such that $|\mathcal{F}_{i_0}| = |\mathcal{F}_{i_0+1}|$ (if there are many such i_0, we pick the minimal one). This implies that every path in $\mathcal{Q}_{i_0}$ intersects C_{i_0+1}.

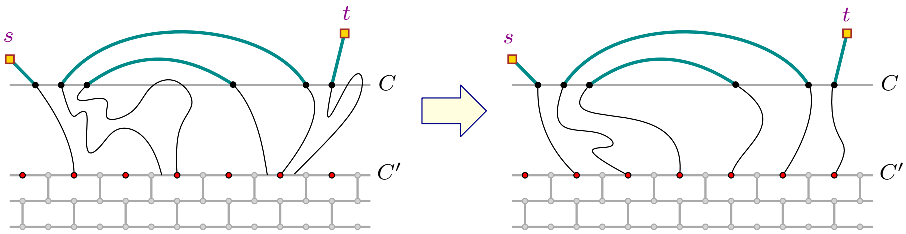

FIG. 6. *A visualization of the statement of Claim* 3.1. *In both figures, the edges* $\{v_F, u_F\}$ *are depicted in blue, the black vertices correspond to the set* Y, *and the red vertices correspond to the set* B. *In the left figure, we illustrate* $|Y|$ *disjoint paths from* Y *to* C', *while in the right figure, we illustrate* $|Y|$ *disjoint paths from* Y *to* B.

For each $F \in \mathcal{F}_{i_0}$, we denote by v_F and u_F the endpoints of F. We define the graph G' obtained from G after removing the internal vertices of every $F \in \mathcal{F}_{i_0}$ (i.e., the vertex set $\bigcup_{F\in\mathcal{F}_{i_0}}(V(F) \setminus \{v_F, u_F\})$) and adding the edge $\{v_F, u_F\}$ for every $F \in \mathcal{F}_{i_0}$. Observe that G' is also planar and contains D_{i_0} as a subgraph. Moreover, notice that, for every $F \in \mathcal{F}_{i_0}$, $\{v_F, u_F\} \in V(C_{i_0}) \cup \{s,t\}$. In Figure 5, $|\mathcal{F}_1| = |\mathcal{F}_2|$ and thus G' is obtained after replacing each 3-colored path with an edge.

In the rest of the proof we will argue that, in G', there is an (s,t)-path that contains all edges $\{v_F, u_F\}$, $F \in \mathcal{F}_{i_0}$, and its intersection with $V(\mathsf{compass}(W^{(h(k))}))$ is the vertex set of a subdivided edge of W that lies in $\mathsf{perim}(W^{(h(k))})$. Having such a path in hand, we can replace each edge $\{v_F, u_F\}$, $F \in \mathcal{F}_{i_0}$ with the corresponding path F and thus obtain the path $\tilde{P}$ claimed in the statement of the lemma.

We will denote by C the cycle C_{i_0} (that is the layer $L_{2k\cdot(i_0-1)+1}$) and by C' the layer $L_{2k\cdot i_0}$. To get some intuition, recall that $C_{i_0+1} = L_{2k\cdot i_0+1}$ and therefore C' is the layer of W "preceding" C_{i_0+1}. Since every path in $\mathcal{Q}_{i_0}$ intersects C_{i_0+1}, it holds that every path in $\mathcal{Q}_{i_0}$ intersects C' at least twice. Therefore, if we set $Y := V(C) \cap \bigcup_{F\in\mathcal{F}_{i_0}}\{v_F, u_F\}$ and $\ell := |Y|$, then $\ell \leq 2k$ and there are ℓ disjoint paths from Y to C' (for an example, see the left part of Figure 6).

Recall that $\mathsf{perim}(W^{(2k\cdot i_0)}) = C'$. We set B to be the set of branch vertices of W that are in $V(C')$ and have degree two in $W^{(2k\cdot i_0)}$. Also, we set $\mathcal{K}$ to be the graph $G' \setminus V(\mathsf{inn}(W^{(2k\cdot i_0)}))$. We now argue that there also exist ℓ disjoint paths from Y to B in $\mathcal{K}$.

CLAIM 3.1. *There is a set* $X \subseteq B$, *a bijection* $\rho : Y \to X$, *and a collection* $\mathcal{P} = \{P_v \mid v \in Y\}$ *of pairwise disjoint paths where, for every* $v \in Y$, P_v *is a* $(v, \rho(v))$-*path in* $\mathcal{K}$.

Proof of Claim 3.1. Suppose, toward a contradiction, that there is a set $S \subseteq V(\mathcal{K})$ of size at most $\ell - 1$ such that there is no path in $\mathcal{K} \setminus S$ from Y to B.

Since there are ℓ disjoint paths from Y to $V(C')$, there is a connected component A of $\mathcal{K} \setminus S$ that contains vertices from both Y and $V(C')$. Since $Y \subseteq V(C)$, A contains vertices from both $V(C)$ and $V(C')$. Also, since $C' = L_{2k\cdot i_0}$, where $i_0 \in [k+2]$ and W has at least $h(k)$ layers, where $h(k) > 2k\cdot(k+2) + 2k$, there exist at least $2k$ vertex disjoint paths from $V(C)$ to B. This, together with the fact that $|S| < \ell$ and $\ell \leq 2k$, implies that there is a connected component A' of $\mathcal{K} \setminus S$ that contains vertices from both $V(C)$ and B.

Since both A and A' contain vertices of both C and C', there exist paths P, P' in A and A', respectively, both intersecting $V(C)$ and $V(C')$. The fact that $C = L_{2k\cdot(i_0-1)+1}$ and $C' = L_{2k\cdot i_0}$ implies that there are $2k$ layers intersecting both $V(P)$

and $V(P')$, that yield $2k$ disjoint paths between $V(P)$ and $V(P')$ in $\mathcal{K}$. Since $|S| < \ell$ and $\ell \leq 2k$, some of the aforementioned disjoint paths between $V(P)$ and $V(P')$ should remain intact in $\mathcal{K} \setminus S$, implying that $A = A'$. But, given that A contains vertices from Y, and A' contains vertices from B, we conclude that S does not separate Y and B, a contradiction to the initial assumption. Therefore, there exist ℓ disjoint paths from Y to B. We set X to be the endpoints (in B) of these paths and this proves the claim. ◻

Following Claim 3.1, let $X \subseteq B$, let a bijection $\rho : Y \to X$, and let there be a collection $\mathcal{P} = \{P_v \mid v \in Y\}$ of pairwise disjoint paths such that for every $v \in Y$, P_v is a $(v, \rho(v))$-path in $\mathcal{K}$.

Now, for each $F \in \mathcal{F}_{i_0}$, we consider the path P_F obtained after joining the paths P_{v_F} and P_{u_F} by the edge $\{v_F, u_F\}$ (in the case where $s, t \in \{v_F, u_F\}$, we just extend the corresponding path in $\mathcal{P}$ by adding the edge $\{v_F, u_F\}$). Let G'' be the graph obtained from G' after contracting each $P_F, F \in \mathcal{F}_{i_0}$ to an edge e_{P_F} and let $E = \{e_{P_F} \mid F \in \mathcal{F}_{i_0}\}$. Then, notice that G'' contains $W^{(2k \cdot i_0)}$ as a subgraph and since $h(k) = 2k \cdot (k+2) + 2k + 1$, the wall $W^{(2k \cdot i_0)}$ has at least $k+1$ layers and therefore height at least $2k+3$. Therefore, by Lemma 3.1, G'' contains an (s,t)-path that contains all edges in E and its intersection with $\mathsf{compass}(W^{(2k \cdot i_0 + k + 1)})$ is a path of $\mathsf{perim}(W^{(2k \cdot i_0 + k + 1)})$ whose endpoints are branch vertices of W.

Thus, using this (s,t)-path in G'', we can find an (s,t)-path $P^\star$ in G that contains S and its intersection with $\mathsf{compass}(W^{(2k \cdot i_0 + k + 1)})$ is a path $\widehat{P}$ of $\mathsf{perim}(W^{(2k \cdot i_0 + k + 1)})$ whose endpoints, say, x and y, are branch vertices of W. Finally, let $R_{x,y}$ be an (x,y)-path in $\mathsf{compass}(W^{(2k \cdot i_0 + k + 1)})$ with the following property: its intersection with $\mathsf{compass}(W^{(h(k))})$ is a path of $\mathsf{perim}(W^{(h(k))})$ whose endpoints are branch vertices of W. The proof concludes by observing if we replace in $P^\star$ the path $\widehat{P}$ with the path $R_{x,y}$, then we obtain an (s,t)-path as claimed in the statement of the lemma. ◻

We stress that, while Lemma 3.2 deals with the case of "rerouting" an (s,t)-path, we can apply the same arguments to "reroute" a cycle that contains a fixed set S away from the inner part of some wall.

3.2. Equivalent instances of small treewidth. In this subsection, we prove that there is an algorithm that receives a framework (G, M), where G is a planar graph of "big enough" treewidth, and two vertices $s, t \in V(G)$, and outputs either a report that G contains an (s,t)-path of rank at least k, or an irrelevant vertex that can be safely removed. In frameworks, to remove a vertex, one has to remove this vertex from G and also restrict the matroid.

Restrictions of matroids. Let $M = (V, \mathcal{I})$ be a matroid and let $S \subseteq V$. We define the *restriction of M to S*, denoted by $M|S$, to be the matroid on the set S whose independent sets are the sets in $\mathcal{I}$ that are subsets of S. Given a $v \in V$, we denote by $M \setminus v$ the matroid $M|(V \setminus \{v\})$.

The goal of this subsection is to prove the following.

LEMMA 3.3. *There is a function $g : \mathbb{N} \to \mathbb{N}$ and an algorithm that, given an integer $k \in \mathbb{N}$, a framework (G, M), where M is a matroid for which we can verify independence in time $\|M\|^{\mathcal{O}(1)}$, and G is a planar graph of treewidth at least $g(k)$, and two vertices $s, t \in V(G)$, outputs, in time $2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$,*

- *either a report that G contains an (s,t)-path of rank at least k or*
- *a vertex $v \in V(G)$ such that (G, M, k, s, t) and $(G \setminus v, M \setminus v, k, s, t)$ are equivalent instances of* MAX RANK (s,t)-PATH.

Moreover, $g(k) = 2^{\mathcal{O}(k \log k)}$.

Keep in mind that if M is represented over a finite field or $\mathbb{Q}$, we can verify independence in time that is a polynomial in $\|M\|$. In order to prove Lemma 3.3, we need some additional definitions and results.

Packings of walls. Let G be a planar graph and W be a wall of G. Let $z, r \in \mathbb{N}$ and q be a nonnegative odd integer. We say that W *admits an* (z,r,q)*-packing of walls* if W has height at least h for some odd $h \geq 2z$, and there is a collection $\mathcal{W} = \{W_0, W_1, \ldots, W_{r-1}\}$ of subwalls of W, such that for every $i \in [0, r-1]$, W_i is a subwall of W of height at least q such that $V(W_i)$ is a subset of $V(W^{(z+1)})$, and for every $i, j \in [0, r-1]$ with $i \neq j$, $V(\mathsf{compass}(W_i))$ and $V(\mathsf{compass}(W_j))$ are disjoint. We call $\mathcal{W}$ an (z,r,q)-*packing* of W (see Figure 3 for a visualization of a packing of a wall W).

Observation 3.1. Given $z, r \in \mathbb{N}$, an odd integer $q \in \mathbb{N}$, and a planar graph G, every wall W of G of height at least $2z + \lceil \sqrt{r} \rceil \cdot (q+1) + 1$ admits a (z,r,q)-packing.

Let W be a wall of a planar graph. We use $\rho(W)$ to denote $r(V(\mathsf{compass}(W)))$.

LEMMA 3.4. *There is a function* $f : \mathbb{N}^4 \to \mathbb{N}$ *and an algorithm that, given integers* $k, z, r, q \in \mathbb{N}$*, where* q *is odd, a framework* (G, M)*, where* G *is planar and* M *is a matroid for which we can verify independence in time* $\|M\|^{\mathcal{O}(1)}$*, and a wall* W *of* G *of height at least* $f(k,z,r,q)$ *such that* $\rho(W) \leq k$*, outputs, in* $(k+1) \cdot r \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$ *time, a subwall* W' *of* W *of height* h *for some odd* $h \in \mathbb{N}$ *such that* $h \geq 2z$*, and a* (z,r,q)*-packing* $\mathcal{W}$ *of* W' *such that for every* $W_i \in \mathcal{W}$*,* $\rho(W_i) = \rho(W')$*. Moreover,* $f(k,z,r,q) = \mathcal{O}(r^{k/2} \cdot z \cdot q)$*.*

Proof. We define the function $f : \mathbb{N}^4 \to \mathbb{N}$ so that, for every $z, r, q \in \mathbb{N}$, $f(0,z,r,q) = 2z + \lceil \sqrt{r} \rceil \cdot (q+1) + 1$, while for $k \geq 1$, we set $f(k,z,r,q) = 2z + \lceil \sqrt{r} \rceil \cdot (f(k-1,z,r,q) + 1) + 1$. Observe that, since q is odd, $f(k,z,r,q)$ is odd for every $k, z, r \in \mathbb{N}$.

We prove the lemma by induction on k. Clearly, if $k = 0$, then the lemma holds trivially, as, by Observation 3.1, there is a (z,r,q)-packing $\mathcal{W}$ of W and also, given that for each $W_i \in \mathcal{W}$ $\mathsf{compass}(W_i)$ is a subgraph of $\mathsf{compass}(W)$, we have that $0 \leq \rho(W_i) \leq \rho(W) = 0$ and thus $\rho(W_i) = 0$. Then the claim holds for $W' = W$ and $\mathcal{W}$.

Suppose now that $k \geq 1$ and that the lemma holds for smaller values of k. We set $w = f(k-1,z,r,q)$. Since W has height at least $2z + \lceil \sqrt{r} \rceil \cdot (w+1) + 1$, Observation 3.1 implies that W admits a (z,r,w)-packing $\mathcal{W}$. Since, by definition, for every $W_i \in \mathcal{W}$, $V(\mathsf{compass}(W_i))$ is a subset of $V(\mathsf{compass}(W))$, it holds that $\rho(W_i) \leq \rho(W)$ for every $W_i \in \mathcal{W}$. We compute $\rho(W)$ and $\rho(W_i)$, for every $W_i \in \mathcal{W}$. This can be done in $r \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$ time. If there is a $W_i \in \mathcal{W}$ such that $\rho(W_i) < \rho(W)$, then, from the induction hypothesis applied to W_i, we have that there exists a subwall W_i' of W_i of height h for some odd $h \geq 2z$ and a (z,r,q)-packing $\mathcal{W}_i$ of W_i' such that all walls in $\mathcal{W}_i$ have the same rank. The lemma follows by observing that $f(k,z,r,q) = \mathcal{O}(r^{k/2} \cdot z \cdot q)$. ∎

We are now ready to prove Lemma 3.3.

Proof of Theorem 3.3. We set

$$b = h(k), \qquad x = k+1, \qquad z = (k+1) \cdot b,$$
$$q = f(k-1, z, x, 3), \qquad r = \lceil \sqrt{k} \rceil \cdot (q+1) + 3, \qquad g(k) = 36(r+1).$$

We first assume that G is 2-connected. If G is not connected, then we break the problem in subproblems, each one corresponding to a 2-connected component B of G and if the vertices of B are separated from s or t by a cut-vertex v of G, then we consider the problem where v is set to be s or t, respectively.

Since the treewidth of G is at least $g(k) = 36(r+1)$, by Proposition 2.1, there is a $(4r+1)$-wall of G. We then consider an r-wall W of G such that $s, t \notin \mathsf{compass}(W)$ and an $(1,k,q)$-packing $\tilde{\mathcal{W}} = \{\tilde{W}_1, \ldots, \tilde{W}_k\}$ of W. This $(1,k,q)$-packing exists because of the fact that $r = \lceil\sqrt{k}\rceil \cdot (q+1) + 3$ and due to Observation 3.1, and we can find it in $\mathcal{O}(n)$ time. For every $i \in [k]$, we set $K_i := V(\mathsf{compass}(\tilde{W}_i))$. Then, compute the rank of K_i for each $i \in [k]$. This can be done in time $k \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$.

If every K_i has rank at least k, then notice that there is a set $S \subseteq V(G)$ such that $r(S) = k$ and for every $i \in [k]$, $|S \cap K_i| = 1$. To obtain an (s,t)-path P such that $S \subseteq V(P)$, we do the following: We first pick two disjoint paths P_s, P_t from the perimeter of W to s and t, respectively (these exist since G is 2-connected). Let D be the perimeter of W and let s' and t' be the endpoints of P_s and P_t in D. Also, let L_2 be the second layer of W. Observe that, since the compass of a wall is a connected graph, there is also a path $\overline{P}$ in G such that the endpoints, say, x, y, of $\overline{P}$ are in L_2, no internal vertex of $\overline{P}$ is a vertex of L_2, and $S \subseteq V(\overline{P})$. Finally, observe that there exist two disjoint paths $P_{s'x}, P_{t'y}$ in the closed disk bounded by D and L_2 connecting s' with x and t' with y, respectively, and that $P := P_s \cup P_{s'x} \cup \overline{P} \cup P_{t'y} \cup P_t$ is an (s,t)-path such that $S \subseteq V(P)$ (see Figure 2).

Suppose now that there is an $i \in [k]$ such that the rank of K_i is at most $k-1$. Since the corresponding wall $\tilde{W}_i$ has height at least $q = f(k-1, z, x, 3)$, by Lemma 3.4, we can find a subwall W' of $\tilde{W}_i$ of height h, for some odd $h \geq 2z$ and a $(z, k+1, 3)$-packing $\mathcal{W} = \{W_0, W_1, \ldots, W_k\}$ of W', such that for every $i \in [0,k]$, $\rho(W_i) = \rho(W')$. We set v to be a central vertex of W_0.

We now prove that (G, M, k, s, t) and $(G \setminus v, M \setminus v, k, s, t)$ are equivalent instances of Max Rank (s,t)-Path. We show that if (G, M, k, s, t) is a yes-instance, then $(G \setminus v, M \setminus v, k, s, t)$ is also a yes-instance, since the other implication is trivial. If (G, M, k, s, t) is a yes-instance, then there is an independent set of vertices $S = \{u_1, \ldots, u_k\} \subseteq V(G)$ and an (s,t)-path P in G such that $r(S) = k$ and $S \subseteq V(P)$. The fact that $z = (k+1) \cdot b$ implies that there is an $i \in [k+1]$ such that the vertex set $V(\mathsf{compass}(W'^{((i-1)\cdot b+1)}) \setminus V(\mathsf{inn}(W'^{(i \cdot b)})))$, which we denote by D_i, does not intersect S. Let S_{in} be the vertices of S that are contained in $\mathsf{compass}(W'^{(i\cdot b)})$ and let S_{out} be the set $S \setminus S_{\text{in}}$. We will show that there is a set $S' \in \mathcal{I}(M \setminus v)$ and a path P' such that $r(S_{\text{out}} \cup S') \geq k$, $S_{\text{out}} \cup S' \subseteq V(P')$ and $V(P') \subseteq V(G \setminus v)$.

We assume that $v \in V(P)$; otherwise we set $S' := S_{\text{in}}$ and $P' := P$ and the lemma follows. We next apply Lemma 3.2 for the wall $W'^{(i-1)\cdot b+1}$; recall that W' has height h, where $h \geq 2(k+1) \cdot b + 1$ and therefore for every $i \in [k+1]$, $W'^{(i-1)\cdot b+1}$ has at least b layers. By Lemma 3.2 applied to $W'^{(i-1)\cdot b+1}$, we get a path $\tilde{P}$ with the following properties: $S_{\text{out}} \subseteq V(\tilde{P})$ and $V(\tilde{P}) \cap V(\mathsf{compass}(W'^{(i\cdot b)}))$ is the vertex set of a path $\hat{P}$ of W'_0 that lies in $\mathsf{perim}(W'^{(i\cdot b)})$ and whose endpoints are branch vertices of $W'^{(i\cdot b)}$. Let $s_{\hat{P}}$ and $t_{\hat{P}}$ be the endpoints of $\hat{P}$.

We can assume that $\rho(W_i) = \rho(W') > 0$ for every $i \in [0,k]$, since otherwise $S_{\text{in}} = \emptyset$ and the claim holds trivially. For every $i \in [0,k]$, since $\rho(W_i) = \rho(W')$ and S_{in} is an independent set of M that is a subset of $\mathsf{compass}(W')$, there is an independent set $S_i \subseteq V(\mathsf{compass}(W_i))$ such that $|S_i| = |S_{\text{in}}|$. Furthermore, because $\rho(W_i) = \rho(W')$ for $i \in [0,k]$, we can choose a set $S' = \{v_1, \ldots, v_k\}$, where v_i is a vertex in S_i for $i \in [k]$ in such a way that $r(S') = \rho(W')$. Then $r(S_{\text{out}} \cup S') = |S_{\text{out}} \cup S_{\text{in}}| \geq k$. Also, notice that, for every $x, y \in L_z$, there is an (x,y)-path $P^\star$ in $W'^{(z)} \setminus (V(L_z) \setminus \{x,y\})$ that contains S' and avoids v. It is easy to see that there exist two disjoint paths Q_1, Q_2 in $\mathsf{compass}(W_0'^{(i\cdot b)})$ connecting $\{s_{\hat{P}}, t_{\hat{P}}\}$ with $\{x,y\}$ and that these paths can be picked to be internally disjoint from $\hat{P}$ and $P^\star$. Thus, if $\tilde{P}''$ is the graph obtained from

$\tilde{P}'$ after removing all internal vertices of $\hat{P}$, then $\tilde{P}'' \cup Q_1 \cup Q_2 \cup P^\star$ is the claimed (s,t)-path that contains $S' \cup S_{\text{out}}$ and avoids v (see Figure 3). ◻

4. Dynamic programming for instances of small treewidth. In this section, we aim to describe a dynamic programming algorithm that solves Max Rank (s,t)-Path for frameworks (G,M), where G has treewidth at most q and M is a linear matroid.

Lemma 4.1. *Let $\mathbb{F}$ be a finite field or $\mathbb{Q}$. There is an algorithm that, given a framework (G,M), where M is an $\mathbb{F}$-linear matroid and G is a graph, two nonnegative integers k and q, where $k \leq q$, and a tree decomposition of G of width q, outputs, in time $2^{q^{\mathcal{O}(1)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$, a report whether G contains an (s,t)-path of rank at least k or not.*

As explained in the next subsection, the algorithm of Lemma 4.1 works for linear matroids represented over any field in which the field operations can be done efficiently, which in particular includes finite fields or $\mathbb{Q}$.

The section is organized as follows. In subsection 4.1, we define nice tree decompositions, the combinatorial structure on which we will perform the dynamic programming, and representative sets, which are used to efficiently encode partial solutions to tables of the dynamic programming. In subsection 4.2, we define the partial solutions of our problem. Then, in subsection 4.3, we present the dynamic programming algorithm of Lemma 4.1, and in subsection 4.4 we prove its correctness. We conclude this section by giving the proof of Theorem 1.1 (subsection 4.5).

4.1. Nice tree decompositions and representative sets. We start this subsection with the definition of nice tree decompositions.

Nice tree decompositions. Let G be a graph. A tree decomposition $\mathcal{T} = (T, \mathcal{X})$ of G is called a *nice tree decomposition* of G if T is rooted to some leaf r and

- for any leaf $l \in V(T)$, $X_l = \emptyset$ (we call X_l a *leaf node* of $\mathcal{T}$, except from X_r, which we call a *root node*),
- every $t \in V(T)$ has at most two children,
- if t has two children t_1 and t_2, then $X_t = X_{t_1} = X_{t_2}$ and X_t is called a *join node,*
- if t has one child t', then
  - either $X_t = X_{t'} \cup \{v\}$ for some $v \in V(G)$ (we call X_t an *insert node*),
  - or $X_t = X_{t'} \setminus \{v\}$ for some $v \in V(G)$ (we call X_t a *forget node*).

It is known that any tree decomposition of G can be transformed into a nice tree decomposition maintaining the same width in linear time [32]. We use G_t to denote the graph induced by the vertex set $\bigcup_{t'} X_{t'}$, where t' ranges over all descendants of t, including t.

In the rest of the paper, given an instance (G,M,k,s,t) of Max Rank (s,t)-Path and a nice tree decomposition $(T,\mathcal{X})$ of G, we will consider the tree decomposition $(T,\mathcal{X}')$ obtained from $(T,\mathcal{X})$ after adding the vertices s and t in every bag in $\mathcal{X}$. Therefore, the leaf nodes and the root node will be equal to the set $\{s,t\}$.

Let (G,M) be a framework and let $(T,\mathcal{X})$ be a tree decomposition of G. For every $t \in V(T)$ and every $i \in \mathbb{N}$, we define $\mathcal{S}_t^{(i)}$ to be the collection of all sets $S \subseteq V(G_t) \setminus X_t$ that are independent sets of M of size i.

Representative sets. Our algorithms use results obtained by Fomin et al. [18] and Lokshtanov et al. [37].

DEFINITION 4.2 (q-representative set). *Let $M=(V,\mathcal{I})$ be a matroid and let $\mathcal{S}$ be a family of subsets of V. For a positive integer q, a subfamily $\widehat{\mathcal{S}}$ is* q-representative *for $\mathcal{S}$ if the following holds: for every set $Y\subseteq V$ of size at most q, if there is a set $X\in\mathcal{S}$ disjoint from Y with $X\cup Y\in\mathcal{I}$, then there is $\widehat{X}\in\widehat{\mathcal{S}}$ disjoint from Y with $\widehat{X}\cup Y\in\mathcal{I}$.*

We write $\widehat{\mathcal{S}}\subseteq^q_{rep}\mathcal{S}$ to denote that $\widehat{\mathcal{S}}\subseteq\mathcal{S}$ is q-representative for S. It is crucial for us that representative families can be computed efficiently for linear matroids. To state these results, we say that a family of sets $\mathcal{S}$ is a *p-family* for an integer $p\geq 0$ if $|S|=p$ for every $S\in\mathcal{S}$.

THEOREM 4.3 (see [18, Theorem 3.8]). *Let $M=(V,\mathcal{I})$ be a linear matroid and let $\mathcal{S}=\{S_1,\dots,S_t\}$ be a p-family of independent sets. Then there exists $\widehat{\mathcal{S}}\subseteq^q_{rep}\mathcal{S}$ of size at most $\binom{p+q}{p}$. Furthermore, given a representation A of M over a field $\mathbb{F}$, there is a randomized algorithm computing $\widehat{\mathcal{S}}\subseteq^q_{rep}\mathcal{S}$ of size at most $\binom{p+q}{p}$ in $\mathcal{O}(\binom{p+q}{p}tp^{\omega}+t\binom{p+q}{q}^{\omega-1})+\|A\|^{\mathcal{O}(1)}$ operations over $\mathbb{F}$, where ω is the exponent of matrix multiplication.*[1]

Observe that the algorithm in Theorem 4.3 is randomized. This is due to the fact that one of the steps of the algorithm constructs a k-truncation[2] of M for $k=p+q$. A k-truncation can be constructed algorithmically for linear matroids, but for general linear matroids, only a randomized algorithm is known [41]. In [37], Lokshtanov et al. gave a deterministic algorithm for linear matroid represented over any field in which the field operations can be done efficiently. In particular, this includes any finite field and the field of rational numbers. This way, they obtained the following theorem.

THEOREM 4.4 (see [37, Theorem 1.3]). *Let $M=(V,\mathcal{I})$ be a linear matroid of rank r and let $\mathcal{S}=\{S_1,\dots,S_t\}$ be a p-family of independent sets. Let A be an $r\times|V|$-matrix representing M over a field $\mathbb{F}$, and let ω be the exponent of matrix multiplication. Then there are deterministic algorithms computing $\widehat{\mathcal{S}}\subseteq^q_{rep}\mathcal{S}$ as follows:*

- *A family $\widehat{\mathcal{S}}$ of size at most $\binom{p+q}{p}$ in $\mathcal{O}(\binom{p+q}{p}^2tp^3r^2+t\binom{p+q}{q}^{\omega}rp)+(r+|V|)^{\mathcal{O}(1)}$ operations over $\mathbb{F}$.*
- *A family $\widehat{\mathcal{S}}$ of size at most $rp\binom{p+q}{p}$ in $\mathcal{O}(\binom{p+q}{p}tp^3r^2+t\binom{p+q}{q}^{\omega-1}(rp)^{\omega-1})+(r+|V|)^{\mathcal{O}(1)}$ operations over $\mathbb{F}$.*

4.2. Partial solutions. We start by defining the notion of *semimatching,* that intuitively encode parts of a path.

Semimatchings. Let X be a set. Letting H be a graph whose vertex set is X, every vertex has degree at most two, and it is acyclic. The collection $\mathcal{M}$ of the edges and the isolated vertices of H is called a *semimatching* of X. Given a semimatching $\mathcal{M}$ of a set X, we use $U(\mathcal{M})$ to denote X. Observe that $|\{\mathcal{M}\mid\mathcal{M}\text{ is a semimatching of }X\}|=2^{\mathcal{O}(|X|\log|X|)}$. We denote by $\mathcal{M}_1^{(v)}$ the set $\{\{v\}\}\cap\mathcal{M}$, by $\mathcal{M}_2^{(v)}$ the set $\{\{u,v\}\mid\{u,v\}\in\mathcal{M}\}$, and by $\mathcal{M}^{(v)}$ the set $\mathcal{M}_1^{(v)}\cup\mathcal{M}_2^{(v)}$. Notice that $|\mathcal{M}_2^{(v)}|\leq 2$.

Given a semimatching $\mathcal{M}$ of a set X and a $v\in X$, we denote by $\mathsf{rem}(\mathcal{M},v)$ the semimatching $\mathcal{M}'=(\mathcal{M}\setminus\mathcal{M}^{(v)})\cup\{\{u\}\mid\{u,v\}\in\mathcal{M}^{(v)}\text{ and }u\notin U(\mathcal{M}\setminus\mathcal{M}^{(v)})\}$. Also, given a set Y such that $X\subsetneq Y$ and a $u\in Y\setminus X$, we denote by $\mathsf{add}(\mathcal{M},u)$ the collection of all semimatchings $\mathcal{M}'$ of $X\cup\{u\}$ such that $\mathcal{M}=\mathsf{rem}(\mathcal{M}',u)$.

[1] The currently best value is $\omega\approx 2.3728596$ [1].

[2] A matroid $M'=(V,\mathcal{I}')$ is a *k-truncation* of $M=(V,\mathcal{I})$ if for every $X\subseteq V$, $X\in\mathcal{I}'$ if and only if $X\in\mathcal{I}$ and $|X|\leq k$.

Linear forests. We say that a graph F is a *linear forest* if it is an acyclic graph of maximum degree two. Let G be a graph and let F be a linear forest that is a subgraph of G. Given a set $X \subseteq V(G)$, we define $\mathsf{sig}_{G,X}(F)$ to be the set that contains (i) all vertices in $X \cap V(F)$ that have degree zero in F and (ii) all pairs $\{u,v\}$ of vertices in $X \cap V(F)$ such that either $\{u,v\} \in E(F)$ or there is a (u,v)-path in F that intersects X only at its endpoints. Notice that $\mathsf{sig}_{G,X}(F)$ is a semimatching of $X \cap V(F)$.

We are now ready to define what is considered as a partial solution to our problem.

Partial solutions. Let G be a graph, let $s,t \in V(G)$, and let $\mathcal{T} = (T,\mathcal{X})$ be a nice tree decomposition of G. Given a $t \in V(T)$, we define a *partial solution* at t to be a quadraple $(X,\mathcal{M},i,S)$, where $\{s,t\} \subseteq X \subseteq X_t$, $\mathcal{M}$ is a semimatching of X, $i \in [k]$, and $S \in \mathcal{S}_t^{(i)}$, such that there is a linear forest $F \subseteq G_t$ where $X = V(F) \cap X_t$, $\mathcal{M} = \mathsf{sig}_{G_t,X_t}(F)$, and $S \subseteq V(F)$. Keep in mind that $S \subseteq V(G_t \setminus X_t)$ and therefore $S \subseteq V(F \setminus X_t)$. We also say that the linear forest F *certifies* that $(X,\mathcal{M},i,S)$ is a partial solution at t. We denote by $\mathcal{B}_t$ the set of all partial solutions at t.

We can easily observe the following.

Observation 4.1. Let (G,M) be a framework and $k \in \mathbb{N}$. Then G contains an (s,t)-path of rank at least k if and only if there is a set $S \in \mathcal{S}_r^{(k)}$ such that $(\{s,t\},\{\{s,t\}\},k,S) \in \mathcal{B}_r$.

4.3. A dynamic programming algorithm. We are now ready to describe the dynamic programming algorithm of Lemma 4.1. For every $t \in V(T)$, we aim to construct a collection $\mathcal{F}_t \subseteq \mathcal{B}_t$ of partial solutions whose size is "small" since we cannot afford to store all independent sets of size i and therefore all partial solutions in $\mathcal{B}_t$. For this reason, we will use representative sets, instead of all possible independent sets using Theorem 4.4, and thus, for every $X \subseteq X_t$, every semimatching $\mathcal{M}$ of X, and every $i \in [k]$, we will keep only a "representative" collection of independent sets $\widehat{\mathcal{S}} \subseteq \mathcal{S}_t^{(i)}$ such that for every $S \in \mathcal{S}_t^{(i)}$, there is a $S' \in \widehat{\mathcal{S}}$ such that $(X,\mathcal{M},i,S) \in \mathcal{B}_t$ if and only if $(X,\mathcal{M},i,S') \in \mathcal{F}_t$. Given a p-family $\mathcal{S}$ of independent sets of a matroid M, we use $\mathsf{Rep}(\mathcal{S})$ to denote the k-representative subfamily $\widehat{\mathcal{S}}$ for $\mathcal{S}$ given by Theorem 4.4.

Leaf node t. Here, as $X_t = \{s,t\}$, the graph $G_t \setminus X_t$ is empty and therefore we set $\mathcal{F}_t = \{(\{s,t\},\{\{s\},\{t\}\},0,\emptyset)\}$.

Insert node t with child t'. We know that $X_t \supseteq X_{t'}$ and $|X_t| = |X_{t'}|+1$. Let v be the vertex in $X_t \setminus X_{t'}$. For every $X \subseteq X_t$ that contains s and t, every semimatching $\mathcal{M}$ of X, and every $i \in [0,k]$, we set

$$\mathcal{S}_t[X,\mathcal{M},i] = \begin{cases} \{S \mid (X,\mathcal{M},i,S) \in \mathcal{F}_{t'}\} & \text{if } v \notin X, \\ \{S \mid (X \setminus \{v\}, \mathsf{rem}(\mathcal{M},v), i, S) \in \mathcal{F}_{t'}\} & \text{if } v \in X \text{ and } \mathcal{M}_2^{(v)} \subseteq E(G_t), \\ \emptyset & \text{if otherwise.} \end{cases}$$

We set $\mathcal{F}_t = \{(X,\mathcal{M},i,S) \mid S \in \mathsf{Rep}(\mathcal{S}_t[X,\mathcal{M},i])\}$.

Forget node t with child t'. We know that $X_t \subseteq X_{t'}$ and $|X_t| = |X_{t'}|-1$. Let v be the vertex in $X_{t'} \setminus X_t$. For every $X \subseteq X_t$ that contains s and t, every semimatching $\mathcal{M}$ of X, and every $i \in [0,k]$, we set

$$\begin{aligned} \mathcal{S}_t[X,\mathcal{M},i] = \{S \mid (X,\mathcal{M},i,S) \in \mathcal{F}_{t'}\} \\ \cup \ \big\{S \mid \exists \mathcal{M}' \in \mathsf{add}(\mathcal{M},v) : (X \cup \{v\}, \mathcal{M}', i, S) \in \mathcal{F}_{t'}\big\} \\ \cup \ \big\{S \cup \{v\} \mid \exists \mathcal{M}' \in \mathsf{add}(\mathcal{M},v) : (X \cup \{v\}, \mathcal{M}', i-1, S) \in \mathcal{F}_{t'} \text{ and} \\ S \cup \{v\} \in \mathcal{I}(M)\big\}. \end{aligned}$$

We set $\mathcal{F}_t = \{(X,\mathcal{M},i,S) \mid S \in \mathsf{Rep}(\mathcal{S}_t[X,\mathcal{M},i])\}$.

Join node t with children t_1 and t_2. We know that $X_t = X_{t_1} = X_{t_2}$. Given a semimatching $\mathcal{M}$ of a set X, we denote by $\xi(\mathcal{M})$ the set of all pairs $(\mathcal{M}_1, \mathcal{M}_2)$ such that $\mathcal{M}_1 \cup \mathcal{M}_2 = \mathcal{M}$ and $\mathcal{M}_1 \cap \mathcal{M}_2 = \emptyset$. For every $X \subseteq X_t$ that contains s and t, every semimatching $\mathcal{M}$ of X, and every $i \in [0,k]$, we set

$$\begin{aligned}\mathcal{S}_t[X,\mathcal{M},i] = \{S_1 \cup S_2 \mid \exists(\mathcal{M}_1,\mathcal{M}_2) \in \xi(\mathcal{M})\ \exists i_1,i_2 \in [0,k], \exists S_1,S_2 \in 2^{V(G)} \cap \mathcal{I}(M):\\ i_1 + i_2 = i, S_1 \cup S_2 \in \mathcal{I}(M), \text{ and}\\ (X_1,\mathcal{M}_1,i_1,S_1) \in \mathcal{F}_{t_1} \text{ and } (X_2,\mathcal{M}_2,i_2,S_2) \in \mathcal{F}_{t_2},\\ \text{where } X_i = U(\mathcal{M}_i),\ i \in [2]\}.\end{aligned}$$

We set $\mathcal{F}_t = \{(X,\mathcal{M},i,S) \mid S \in \mathsf{Rep}(\mathcal{S}_t[X,\mathcal{M},i])\}$.

Our dynamic programming algorithm computes $\mathcal{F}_t$ for every $t \in V(T)$ in a bottom-up manner and checks whether there is a set $S \in 2^{V(G)} \cap \mathcal{I}(M)$ of size k such that $(\{s,t\},\{\{s,t\}\},k,S) \in \mathcal{F}_r$. If so, it outputs a report that there is an (s,t)-path of G of rank at least k; otherwise it outputs a report that such a path does not exist.

4.4. Proof of correctness of the dynamic programming algorithm. To prove the correctness of the algorithm presented in subsection 4.3, we first prove the following.

LEMMA 4.5. *For every $t \in V(T)$, $\mathcal{F}_t \subseteq \{(X,\mathcal{M},i,S) \mid S \in \mathcal{S}_t[X,\mathcal{M},i]\} \subseteq \mathcal{B}_t$ and $|\mathcal{F}_t| = 2^{q^{\mathcal{O}(1)}}$.*

Proof. We prove the lemma by bottom-up induction on the decomposition tree. Let $t \in V(T)$. We distinguish cases depending on the type of node X_t.

Case 1: X_t is a leaf node.

In this case, the statement of the lemma holds trivially.

In the following cases (i.e., when X_t is either an insert node, a forget node, or a join node), it suffices to show that $\{(X,\mathcal{M},i,S) \mid S \in \mathcal{S}_t[X,\mathcal{M},i]\} \subseteq \mathcal{B}_t$, since, due to Theorem 4.4, $\mathcal{F}_t \subseteq \{(X,\mathcal{M},i,S) \mid S \in \mathcal{S}_t[X,\mathcal{M},i]\}$ and $|\mathcal{F}_t| = 2^{q^{\mathcal{O}(1)}}$. Recall that a tuple (X,M,i,S) belongs to $\mathcal{B}_t$ if there is a linear forest $F \subseteq G_t$, where $X = V(F) \cap X_t$, $\mathcal{M} = \mathsf{sig}_{G_t,X_t}(F)$, and $S \subseteq V(F)$, while also it holds that $S \in \mathcal{S}_t^{(i)}$.

Case 2: X_t is an insert node.

Let t' be the child of t. Let $(X,\mathcal{M},i,S)$ such that $S \in \mathcal{S}_t[X,\mathcal{M},i]$. If $v \notin X$, then $(X,\mathcal{M},i,S) \in \mathcal{F}_{t'}$. By the induction hypothesis, there is a linear forest F' that certifies that $(X,\mathcal{M},i,S) \in \mathcal{B}_{t'}$. Since $v \notin X$ and $X_t = X_{t'} \cup \{v\}$, F' is also a linear forest in G_t where $X = V(F') \cap X_t$ and $\mathcal{M} = \mathsf{sig}_{G_t,X_t}(F')$. Therefore, F' certifies that $(X,\mathcal{M},i,S) \in \mathcal{B}_t$. If $v \in X$ and $\mathcal{M}_2^{(v)} \subseteq E(G_t)$, then $(X \setminus \{v\}, \mathsf{rem}(\mathcal{M},v), i, S) \in \mathcal{F}_{t'}$ and therefore, by the induction hypothesis, $(X \setminus \{v\}, \mathsf{rem}(\mathcal{M},v), i, S) \in \mathcal{B}_{t'}$. This implies that there is a linear forest F' certifying that $(X \setminus \{v\}, \mathsf{rem}(\mathcal{M},v), i, S) \in \mathcal{B}_{t'}$. Since $\mathsf{sig}_{G_{t'},X_{t'}}(F') = \mathsf{rem}(\mathcal{M},v)$, $\mathcal{M}$ is a semimatching of X, and $\mathcal{M}_2^{(v)} \subseteq E(G_t)$, we have that $F' \cup \{v, \mathcal{M}_2^{(v)}\}$ is a linear forest, which we denote by F. Observe that F certifies that $(X,\mathcal{M},i,S) \in \mathcal{B}_t$.

Case 3: X_t is a forget node.

Let t' be the child of t and let $(X,\mathcal{M},i,S)$ such that $S \in \mathcal{S}_t[X,\mathcal{M},i]$. Observe that if $(X,\mathcal{M},i,S) \in \mathcal{F}_{t'}$, there is a linear forest F' certifying that $(X,\mathcal{M},i,S) \in \mathcal{B}_{t'}$ and the fact that $v \notin X_t$ (and therefore $v \notin X$) implies that F' also certifies that $(X,\mathcal{M},i,S) \in \mathcal{B}_t$. If there is an $\mathcal{M}' \in \mathsf{add}(\mathcal{M},v)$ such that $(X \cup \{v\}, \mathcal{M}', i, S) \in \mathcal{F}_{t'}$, then there is a linear forest F' certifying that $(X \cup \{v\}, \mathcal{M}', i, S) \in \mathcal{B}_{t'}$. In this case, $V(F') \cap X_t = X$ and $\mathsf{sig}_{G_t,X_t}(F') = \mathcal{M}$. Therefore, F' certifies that $(X,\mathcal{M},i,S) \in \mathcal{B}_t$. Finally, if $S = S' \cup \{v\}$ and there is an $\mathcal{M}' \in \mathsf{add}(\mathcal{M},v)$ such that $(X \cup \{v\}, \mathcal{M}', i-1, S') \in \mathcal{F}_{t'}$ and

$S' \cup \{v\} \in \mathcal{I}(M)$, there is a linear forest F' that certifies that $(X \cup \{v\}, \mathcal{M}', i-1, S') \in \mathcal{B}_{t'}$. The same linear forest F' certifies that $(X, \mathcal{M}, i, S) \in \mathcal{B}_t$.

Case 4: X_t is a join node.

Let t_1 and t_2 be the two children of t and let $(X, \mathcal{M}, i, S)$ such that $S \in \mathcal{S}_t[X, \mathcal{M}, i]$. By definition, there exist a pair $(\mathcal{M}_1, \mathcal{M}_2) \in \xi(\mathcal{M})$, two integers $i_1, i_2 \in [0, k]$, and sets $S_1, S_2 \in 2^{V(G)} \cap \mathcal{I}(M)$ such that $i_1 + i_2 = i$, $S = S_1 \cup S_2 \in \mathcal{I}(M)$, and $(X_1, \mathcal{M}_1, i_1, S_1) \in \mathcal{F}_{t_1}$, $(X_2, \mathcal{M}_2, i_2, S_2) \in \mathcal{F}_{t_2}$, where $X_i = U(\mathcal{M}_i), i \in [2]$. By the induction hypothesis, there is a linear forest $F_1 \subseteq G_{t_1}$ certifying that $(X_1, \mathcal{M}_1, i_1, S_1) \in \mathcal{B}_{t_1}$ and a linear forest $F_2 \subseteq G_{t_2}$ certifying that $(X_2, \mathcal{M}_2, i_2, S_2) \in \mathcal{B}_{t_2}$. Since $\mathcal{M}_1 \cup \mathcal{M}_2 = \mathcal{M}$ and $\mathcal{M}_1 \cap \mathcal{M}_2 = \emptyset$, it holds that $F_1 \cup F_2$ is a linear forest of G_t such that $\mathsf{sig}_{G_t, X_t}(F_1 \cup F_2) = \mathcal{M}$. Therefore, we get that $F_1 \cup F_2$ certifies that $(X, \mathcal{M}, i, S) \in \mathcal{B}_t$. ∎

We now show the following lemma, which is an intermediate step toward the proof of correctness of our dynamic programming algorithm.

LEMMA 4.6. *Let P be an (s,t)-path of G and let S be a subset of $V(P)$ that is an independent set of M of size at least k. For every $t \in V(T)$, if F_t is the graph $P \cap G_t$, $S_t = S \cap V(G_t \setminus X_t)$, $|S_t| = i$, and $X = V(P) \cap X_t$, and $\mathcal{M} = \mathsf{sig}_{G_t, X_t}(F_t)$, then for every $S' \in \mathcal{S}_t[X, \mathcal{M}, i]$, if F' certifies that $(X, \mathcal{M}, i, S') \in \mathcal{B}_t$, then $F' \cup (P \setminus \mathsf{int}(V(F_t)))$, where $\mathsf{int}(F_t)$ denotes the vertices of F_t in $V(G_t) \setminus X_t$, is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_t)$.*

Proof. We prove the lemma by bottom-up induction on the decomposition tree. Let P be an (s,t)-path of G and let S be a subset of $V(P)$ that is an independent set of M of size at least k. Also, let $t \in V(T)$. Let F_t be the graph $P \cap G_t$, $S_t = S \cap V(G_t \setminus X_t)$, $|S_t| = i$, and $X = V(P) \cap X_t$, and $\mathcal{M} = \mathsf{sig}_{G_t, X_t}(F_t)$.

Case 1: X_t is an leaf node.

In this case, the lemma holds trivially.

Case 2: X_t is an insert node.

Let t' be the child of t. By the induction hypothesis, if $F_{t'}$ is the graph $P \cap G_{t'}$, $S_{t'} = S \cap V(G_{t'} \setminus X_{t'})$, $|S_{t'}| = i$, $X' = V(P) \cap X_{t'}$, and $\mathcal{M}' = \mathsf{sig}_{G_{t'}, X_{t'}}(F_{t'})$, then if $S'' \in \mathcal{S}_{t'}[X', \mathcal{M}', i]$ such that $(X', \mathcal{M}', i, S'') \in \mathcal{B}_{t'}$ and F'' certifies that $(X', \mathcal{M}', i, S'') \in \mathcal{B}_{t'}$ then $F'' \cup (P \setminus \mathsf{int}(F_{t'}))$ is an (s,t)-path of G that contains the set $S'' \cup (S \setminus S_{t'})$. We will prove that for every $S' \in \mathcal{S}_t[X, \mathcal{M}, i]$, if F' certifies that $(X, \mathcal{M}, i, S') \in \mathcal{B}_t$, then $F' \cup (P \setminus \mathsf{int}(F_t))$ is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_t)$.

If $v \notin X$, then the fact that $X_t = X_{t'} \cup \{v\}$ implies that $F_t = F_{t'}$, $S_t = S_{t'}$, $X = X'$, and $\mathcal{M} = \mathcal{M}'$. Therefore, since in this case $\mathcal{S}_t[X, \mathcal{M}, i] = \{S \mid (X, \mathcal{M}, i, S) \in \mathcal{F}_{t'}\}$, it holds that $(X, \mathcal{M}, i, S') \in \mathcal{F}_{t'}$. Also, by Lemma 4.5, $\mathcal{F}_{t'} \subseteq \{(X, \mathcal{M}, i, S) \mid S \in \mathcal{S}_{t'}[X, \mathcal{M}, i]\} \subseteq \mathcal{B}_{t'}$. Therefore, if F' certifies that $(X, \mathcal{M}, i, S') \in \mathcal{B}_{t'}$, then $F' \cup (P \setminus \mathsf{int}(F_{t'}))$ is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_{t'})$. Observe that F' also certifies that $(X, \mathcal{M}, i, S') \in \mathcal{B}_t$ and since $F_t = F_{t'}$ and $S_t = S_{t'}$, we have $F' \cup (P \setminus \mathsf{int}(F_t)) = F' \cup (P \setminus \mathsf{int}(F_{t'}))$ and $S' \cup (S \setminus S_t) = S' \cup (S \setminus S_{t'})$. Thus, $F' \cup (P \setminus \mathsf{int}(F_t))$ is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_t)$.

If $v \in X$ and $\mathcal{M}_2^{(v)} \subseteq E(G_t)$, then observe that $F_{t'} = F_t \setminus \{v\}$, $S_t = S_{t'}$ (since $S_t = S \cap V(G_t \setminus X_t) = S \cap V(G_{t'} \setminus X_{t'})$), $X' = X \setminus \{v\}$, and $\mathcal{M}' = \mathsf{rem}(\mathcal{M}, v)$. Therefore, since $\mathcal{S}_t[X, \mathcal{M}, i] = \{S \mid (X \setminus \{v\}, \mathsf{rem}(\mathcal{M}, v), i, S) \in \mathcal{F}_{t'}\}$, and $S' \in \mathcal{S}_t[X, \mathcal{M}, i]$, we have that $(X', \mathcal{M}', i, S') \in \mathcal{F}_{t'}$. Also, by Lemma 4.5, $\mathcal{F}_{t'} \subseteq \{(X, \mathcal{M}, i, S) \mid S \in \mathcal{S}_{t'}[X, \mathcal{M}, i]\} \subseteq \mathcal{B}_{t'}$. Therefore, if F'' certifies that $(X', \mathcal{M}', i, S') \in \mathcal{B}_{t'}$, then $F'' \cup (P \setminus \mathsf{int}(F_{t'}))$ is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_{t'})$. Let $F' = F'' \cup (\{v\}, \mathcal{M}_2^{(v)})$. Notice that F' is a linear forest and this follows from the fact that F_t and F'' are linear forests, $F_{t'} = F_t \setminus \{v\}$, $\mathsf{sig}_{G_{t'}, X_{t'}}(F_{t'}) = \mathsf{sig}_{G_{t'}, X_{t'}}(F'')$, and $\mathcal{M}_2^{(v)} \subseteq E(G_t)$. Therefore, F'

certifies that $(X, \mathcal{M}, i, S'') \in \mathcal{B}_t$. Also, we have that $F' \cup (P \setminus V(F_t)) = F'' \cup (P \setminus \mathsf{int}(F_{t'}))$ and $S'' \cup (S \setminus S_t) = S'' \cup (S \setminus S_{t'})$. Thus, $F' \cup (P \setminus \mathsf{int}(F_t))$ is an (s,t)-path of G that contains the set $S'' \cup (S \setminus S_t)$. To conclude Case 2, observe that if $v \in X$ and $E(G_t) \setminus \mathcal{M}_2^{(v)} \neq \emptyset$, $\mathcal{S}_t[X, \mathcal{M}, i] = \emptyset$.

Case 3: X_t is a forget node.

Let t' be the child of t and let v be the vertex in $X_{t'} \setminus X_t$. By the induction hypothesis, if $F_{t'}$ is the graph $P \cap G_{t'}$, $S_{t'} = S \cap V(G_{t'} \setminus X_{t'})$, $|S_{t'}| = i$, $X' = V(P) \cap X_{t'}$, and $\mathcal{M}' = \mathsf{sig}_{G_{t'}, X_{t'}}(F_{t'})$, then if $S'' \in \mathcal{S}_{t'}[X', \mathcal{M}', i]$ and F'' certifies that $(X', \mathcal{M}', i, S'') \in \mathcal{B}_{t'}$ then $F'' \cup (P \setminus \mathsf{int}(F_{t'}))$ is an (s,t)-path of G that contains the set $S'' \cup (S \setminus S_{t'})$. Let $S' \in \mathcal{S}_t[X, \mathcal{M}, i]$.

If $(X, \mathcal{M}, i, S') \in \mathcal{F}_{t'}$, then, by Lemma 4.5, $(X, \mathcal{M}, i, S') \in \mathcal{B}_{t'}$, and therefore there is a linear forest $F'' \subseteq G_{t'}$ certifying that $(X, \mathcal{M}, i, S') \in \mathcal{B}_{t'}$. Observe that since $V(F'') \cap X_{t'} = X$, we have that F'' is also a linear forest in G_t certifying that $(X, \mathcal{M}, i, S') \in \mathcal{B}_t$. Therefore, since $F'' \cup (P \setminus \mathsf{int}(F_{t'})) = F'' \cup (P \setminus \mathsf{int}(F_t))$ and $S'' \cup (S \setminus S_t) = S'' \cup (S \setminus S_{t'})$, we have that $F'' \cup (P \setminus \mathsf{int}(F_t))$ is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_t)$.

If there is an $\mathcal{M}' \in \mathsf{add}(\mathcal{M}, v)$ such that $(X \cup \{v\}, \mathcal{M}', i, S') \in \mathcal{F}_{t'}$, then, by Lemma 4.5, $(X \cup \{v\}, \mathcal{M}', i, S') \in \mathcal{B}_{t'}$ and therefore there is a linear forest $F'' \subseteq G_{t'}$ certifying that $(X \cup \{v\}, \mathcal{M}', i, S') \in \mathcal{B}_{t'}$. Notice that, since $X_t = X_{t'} \setminus \{v\}$, we have that $V(F'') \cap X_t$ and $\mathsf{sig}_{G_t, X_t}(F'') = \mathcal{M}$. Thus, F'' certifies that $(X, \mathcal{M}, i, S') \in \mathcal{B}_t$. Moreover, since $F'' \cup (P \setminus \mathsf{int}(F_{t'})) = F'' \cup (P \setminus \mathsf{int}(F_t))$ and $S'' \cup (S \setminus S_t) = S'' \cup (S \setminus S_{t'})$, we have that $F'' \cup (P \setminus \mathsf{int}(F_t))$ is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_t)$.

If $S' = S'' \cup \{v\}$ and there is an $\mathcal{M}' \in \mathsf{add}(\mathcal{M}, v)$ such that $(X \cup \{v\}, \mathcal{M}', i-1, S'') \in \mathcal{F}_{t'}$ and $S' \in \mathcal{I}(M)$, then by Lemma 4.5, $(X \cup \{v\}, \mathcal{M}', i-1, S'') \in \mathcal{B}_{t'}$ and therefore there is a linear forest $F'' \subseteq G_{t'}$ certifying that $(X \cup \{v\}, \mathcal{M}', i-1, S'') \in \mathcal{B}_{t'}$. The fact that $X_t = X_{t'} \setminus \{v\}$ implies that $S' \subseteq V(G_t \setminus X_t)$, $V(F'') \cap X_t$ and $\mathsf{sig}_{G_t, X_t}(F'') = \mathcal{M}$. Therefore, F'' certifies that $(X, \mathcal{M}, i, S') \in \mathcal{B}_t$. Moreover, since $F'' \cup (P \setminus \mathsf{int}(F_{t'})) = F'' \cup (P \setminus \mathsf{int}(F_t))$ and $S'' \cup (S \setminus S_t) = S'' \cup (S \setminus S_{t'})$, we have that $F'' \cup (P \setminus \mathsf{int}(F_t))$ is an (s,t)-path of G that contains the set $S' \cup (S \setminus S_t)$. This concludes Case 3.

Case 4: X_t is a join node.

Let t_1, t_2 be the two children of t and assume that the induction hypothesis holds for both t_1, t_2. Keep in mind that $X_{t_1} = X_{t_2} = X_t$. Also, let $S' \in \mathcal{S}_t[X, \mathcal{M}, i]$. By definition, $S' = S_1 \cup S_2$ such that there is a pair $(\mathcal{M}_1, \mathcal{M}_2) \in \xi(\mathcal{M})$ and two integers $i_1, i_2 \in [0,k]$ such that $i_1 + i_2 = i$, and if $X_i = U(\mathcal{M}_i), i \in [2]$, then $S_1 \cup S_2 \in \mathcal{I}(M)$, $(X_1, \mathcal{M}_1, i_1, S_1) \in \mathcal{F}_{t_1}$, and $(X_2, \mathcal{M}_2, i_2, S_2) \in \mathcal{F}_{t_2}$. Due to Lemma 4.5, $(X_1, \mathcal{M}_1, i_1, S_1) \in \mathcal{B}_{t_1}$ and $(X_2, \mathcal{M}_2, i_2, S_2) \in \mathcal{B}_{t_2}$, and therefore there are linear forests $F_1 \subseteq G_{t_1}$ and $F_2 \subseteq G_{t_2}$ such that for every $i \in [2]$, F_i certifies that $(X_i, \mathcal{M}_i, i_i, S_i) \in \mathcal{B}_{t_i}$. Moreover, by the induction hypothesis, $F_i \cup (P \setminus \mathsf{int}(F_{t_i}))$ is an (s,t)-path of G that contains the set $S_i \cup (S \setminus S_{t_i})$. The fact that $(\mathcal{M}_1, \mathcal{M}_2) \in \xi(\mathcal{M})$ implies that $F_1 \cup F_2$ is a linear forest of G_t such that $V(F_1 \cup F_2) \cap X_t = X$ and $\mathsf{sig}_{G_t, X_t}(F_1 \cup F_2) = \mathcal{M}$. Moreover, since $S_1 \cup S_2 \in \mathcal{I}(M)$ and $i_1 + i_2 = i$, we get that $F_1 \cup F_2$ certifies that $(X, \mathcal{M}, i, S_1 \cup S_2) \in \mathcal{B}_t$. Also, the fact that for every $i \in [2]$, $F_i \cup (P \setminus \mathsf{int}(F_{t_i}))$ is an (s,t)-path of G that contains the set $S_i \cup (S \setminus S_{t_i})$ implies that $F_1 \cup F_2 \cup (P \setminus \mathsf{int}(F_t))$ is an (s,t)-path of G that contains the set $S_1 \cup S_2 \cup (S \setminus S_t)$. This concludes Case 4. ☐

We conclude this subsection by proving Lemma 4.1.

Proof of Theorem 4.1. We first observe that we can transform a given tree decomposition of width q to a nice tree decomposition (T, χ) of width q in time $\mathcal{O}(q^2 \cdot n)$. Moreover, $|V(T)| = q^{\mathcal{O}(1)} \cdot n$. Then, for every $t \in V(T)$, we compute the set $\mathcal{F}_t$ in a bottom-up way. By Lemma 4.5, $|\mathcal{F}_t| = 2^{q^{\mathcal{O}(1)}}$ and each $\mathcal{F}_t$ can be computed in time

$2^{q^{\mathcal{O}(1)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$, resulting in the claimed overall running time. If there is a set $S \subseteq V(G)$ such that $\{\emptyset, \{\emptyset\}, k, S\} \in \mathcal{F}_r$, then we output a report that G contains an (s,t)-path of rank at least k; otherwise we report that such a path does not exist in G. The correctness of the algorithm follows from Observation 4.1, Lemma 4.6, and the fact that, by Theorem 4.4, for every $t \in V(T)$, $\mathcal{F}_t$ is a k-representative family and $\mathcal{F}_t \subseteq \{(X, \mathcal{M}, i, S) \mid S \in \mathcal{S}_t[X, \mathcal{M}, i]\}$. ∎

4.5. Proof of Theorem 1.1. In the proof of Theorem 1.1, we will use the single-exponential time 2-approximation algorithm for the treewidth of Korhonen [34].

PROPOSITION 4.7. *There exists an algorithm that given a graph G and an integer $k \in \mathbb{N}$, outputs, in time $2^{\mathcal{O}(k)} \cdot |G|$, either a tree decomposition of G of width at most $2k+1$ or a report that the treewidth of G is larger than k.*

Proof of Theorem 1.1. Let (G, M) be a framework, where G is a planar graph and M is a linear matroid given by its representation over a finite filed or the field of rationals, and let $k \in \mathbb{N}$. We set $q = g(k)$, where g is the function of Lemma 3.3. Keep in mind that $g(k) = 2^{\mathcal{O}(k \log k)}$. We describe an algorithm $\mathcal{A}$ that solves MAX RANK (s,t)-PATH.

Our algorithm $\mathcal{A}$ first calls the algorithm of Proposition 4.7 for G and q which runs in time $2^q \cdot n = 2^{2^{\mathcal{O}(k \log k)}} \cdot n$ and outputs either a tree decomposition of G of width at most $2q$ or a report that the treewidth of G is larger than q. In the first possible output, we use the algorithm of Lemma 4.1, which runs in time $2^{q^{\mathcal{O}(1)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)} = 2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$, and we solve MAX RANK (s,t)-PATH. In the second possible output (i.e., where G has treewidth at least q), we apply the algorithm of Lemma 3.3 and, in time $2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$, we either report a positive answer to MAX RANK (s,t)-PATH or find a vertex $v \in V(G)$ such that (G, M, k, s, t) and $(G \setminus v, M \setminus v, k, s, t)$ are equivalent instances of MAX RANK (s,t)-PATH. If the latter happens, we recursively run $\mathcal{A}$ for the framework $(G \setminus v, M \setminus v)$. Observe that the overall running time of $\mathcal{A}$ is $2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$. ∎

5. Computational lower bound for MAX RANK (s,t)-PATH. In this section, we prove the unconditional computational lower bound given in Theorem 1.2, which we restate.

THEOREM 1.2. *There is no algorithm solving* MAX RANK (s,t)-PATH *for frameworks with matroids represented by the independence oracles using $f(k) \cdot n^{o(k)}$ oracle calls for any computable function f. Furthermore, the lower bound holds for frameworks with planar graphs of treewidth at most two.*

Proof. Let p and q be positive integers. We construct the framework $(G_{p,q}, M_P)$, where M_P is constructed for a given path P in $G_{p,q}$, as follows.

First, we construct $G_{p,q}$ on $2pq + q + 1$ vertices with two terminals s and t (see Figure 7).

- Construct $q+1$ vertices $w_0, \ldots, w_q$, and set $s = w_0$ and $t = w_q$.
- For each $i \in [q]$, construct $2p$ vertices $u_1^i, \ldots u_p^i$ and $v_1^i, \ldots, v_p^i$, make u_j^i and v_j^i adjacent for all $j \in [p]$, make $u_1^i, \ldots u_p^i$ adjacent to w_{i-1}, and make $v_1^i, \ldots, v_p^i$ adjacent to w_i.

It is easy to see that $G_{p,q}$ is a planar graph of treewidth at most two.

To define M_P, consider an (s,t)-path P in $G_{p,q}$. We also set $W = \bigcup_{i=1}^{q} \bigcup_{j=1}^{p} \{u_j^i, v_j^i\}$. We define $\mathcal{I} \subseteq 2^{V(G_{p,q})}$ to be the set containing all sets $X \subseteq W$ of size at most $2q$ such that

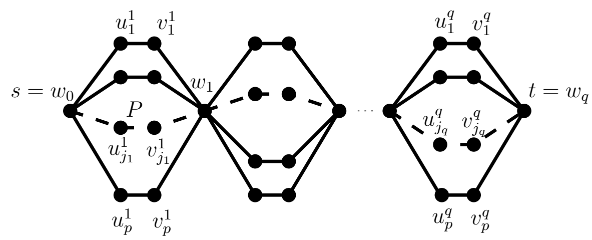

FIG. 7. *Construction of $G_{p,q}$; P is shown by a dashed line.*

- either $|X| < 2q$,
- or $|X| = 2q$ and $X \subseteq V(P)$,
- or $|X| = 2q$ and there is no (s,t)-path Q in $G_{p,q}$ with $X \subseteq V(Q)$.

CLAIM 5.1. *$M_P = (V(G_{p,q}), \mathcal{I})$ is a matroid.*

Proof of Claim 5.1. To show the claim, we have to verify that the independence axioms are fulfilled for $\mathcal{I}$. The definition of $\mathcal{I}$ immediately implies that (I1) and (I2) hold and it remains to verify (I3). For this, we consider $X, Y \in \mathcal{I}$ such that $|X| < |Y|$. Recall that we have to show that there is $x \in Y \setminus X$ such that $X \cup \{x\} \in \mathcal{I}$. This is trivial if $|X| \leq 2q - 2$ because for every $x \in Y \setminus X$, $|X \cup \{x\}| < 2q$, and therefore $X \cup \{x\} \in \mathcal{I}$. Suppose that $|X| = 2q - 1$. If there is no (s,t)-path Q in $G_{p,q}$ such that $X \subseteq V(Q)$, then $X \cup \{x\}$ has the same property for any $x \in Y \setminus X$ and $X \cup \{x\} \in \mathcal{I}$ by the definition of $\mathcal{I}$. Assume that $X \subseteq V(Q)$ for some (s,t)-path Q. By the construction of $G_{p,q}$, for each $i \in [q]$, Q contains exactly two vertices $u_{j_i}^i$ and $v_{j_i}^i$ for some $j_i \in [p]$. Furthermore, because $|X| = 2q - 1$ and $X \subseteq W$, there is unique $i' \in [q]$ such that $|\{u_{j_i}^i, v_{j_i}^i\} \cap X| = 1$ and $u_{j_i}^i, v_{j_i}^i \in X$ for all other $i \in [q]$. This implies that if there is $x \in Y$ such that $x \notin V(Q)$, then $x \in Y \setminus X$ and $X \cup \{x\}$ in not contained in any (s,t)-path. Thus, $X \cup \{x\} \in \mathcal{I}$. From now on, we assume that $Y \subseteq V(Q)$. This means that $Q = P$. We have that $X \subseteq Y$ and $|Y \setminus X| = 1$. For the unique $x \in Y \setminus X$, we obtain that $X \cup \{x\} = Y \in \mathcal{I}$. This concludes the proof. ◻

The proof of Theorem 1.2 is based on the following crucial claim about solving MAX RANK (s,t)-PATH for instances $(G_{p,q}, M_P, 2q)$.

CLAIM 5.2. *Solving* MAX RANK (s,t)-PATH *for instances $(G_{p,q}, M_P, k)$ for $k = 2q$ with the matroids M_P defined by the independence oracle for an (unknown) (s,t)-path P in $G_{p,q}$ demands at least $p^q - 1$ oracle queries.*

Proof of Claim 5.2. Observe that P is a unique (s,t)-path of rank at least $k = 2q$. Thus, the task is to find P which is known only to the oracle. Querying the oracle for $X \subseteq V(G_{p,q})$ such that $X \setminus W \neq \emptyset$ does not provide any information about P because all such sets are not independent. Similarly, querying the oracle for $X \subseteq W$ such that either $|X| < k$ or $|X| > k$ does not help to find P as all sets of size at most $k-1$ are independent and all sets of size at least $k+1$ are not independent. If $X \subseteq W$ of size k is not a subset of vertices of an (s,t)-path, then X is always independent. Thus, we can assume that the oracle is queried only for $X \subseteq W$ of size $k = 2q$ such that $X \subseteq V(Q)$ for some (s,t)-path Q in $G_{p,q}$. Suppose that the oracle is queried for at most $q^p - 2$ sets X of this type and for all these queries the oracle returned that the sets are not independent. Notice that $G_{p,q}$ has p^q distinct (s,t)-paths. Then there are two distinct (s,t)-paths Q, Q' such that the oracle was not queried for $X = V(Q) \cap W$

and $X' = V(Q') \cap W$. The previous queries do not help to distinguish between Q and Q'. Hence, at least one more query is unavoidable. This proves the claim. ◻

Now we are ready to show the claim of Theorem 1.2. Suppose that there is an algorithm $\mathcal{A}$ solving MAX RANK (s,t)-PATH with at most $f(k)\cdot n^{g(k)}$ oracle queries for some computable functions f and g such that $g(k) = o(k)$. Without loss of generality, we assume that f and g are monotone nondecreasing functions. Because $g(k) = o(k)$, there is a positive integer K such that $g(k) < k/8$ for all $k \geq K$. Then for each $k \geq K$, there is a positive integer N_k such that for every $n \geq N_k$, $f(k) \cdot n^{g(k)} < n^{2g(k)} < n^{k/4} \leq \left(\frac{n}{k} - 1\right)^{k/2} - 1$, and therefore $f(k) \cdot n^{g(k)} < \left(\frac{n}{k} - 1\right)^{k/2} - 1$. Consider instances $(G_{p,q}, M_P, k)$ for even $k \geq K$, where $q = k/2$ and $p \geq N_k/k$. Then $k = 2q$ and $n = 2pq + q + 1$. In particular, $n \geq pk \geq N_k$. Then $\mathcal{A}$ would solve the problem with at most $f(k) \cdot n^{g(k)} < \left(\frac{n}{k} - 1\right)^{k/2} - 1 \leq p^q - 1$ oracle queries. However, this would contradict Claim 5.2. This completes the proof. ◻

6. Conclusion. In this paper, we provide a deterministic FPT algorithm for MAX RANK (s,t)-PATH for frameworks (G,M), where G is a planar graph and M is represented over a finite field or the rationals. We complement this result by proving that there is no FPT algorithm for the problem when the input matroids are given by their independence oracles even if the input graphs are planar graphs of treewidth at most two. Let us conclude by discussing some open research directions.

Since the algorithm of [16] for MAX RANK (s,t)-PATH runs in $2^{\mathcal{O}(k^2 \log(k+q))} n^{\mathcal{O}(1)}$ time, a natural question is whether one can drop the double-exponential dependence on the parameter k on the running time of the algorithm of Theorem 1.1. The main bottleneck is the bound the treewidth of a graph that contains no irrelevant vertices. In particular, our approach to detect irrelevant vertices requires a recursive zooming into a given wall of the graph in order to find a packing of $k+1$-many k-walls with compasses of specific rank. To perform this zooming, one should ask for the initial wall to be of height at least $k^{\mathcal{O}(k)}$. It is unclear whether we can circumvent this argument and detect irrelevant vertices if the initial wall has height linear (or even polynomial) in k.

As mentioned in the introduction, the method of [16] gives a randomized algorithm for the more general problem of MAXIMUM RANK (S,T)-LINKAGE. In this paper, we focus on the special case where $|S| = |T| = 1$ and one could ask whether our techniques can be applied to solve the general problem of detecting (S,T)-linkages of large rank for frameworks with planar graphs and matroids represented over finite fields. Such a generalization of our results does not seem to be trivial, and therefore we leave this as an open research direction.

Another natural question to ask is whether our approach can be generalized to obtain deterministic FPT algorithms for frameworks with more general classes of graphs. While it seems plausible to extend the applicability of the irrelevant vertex technique arguments up to graphs that exclude a graph as a minor, such a proof would be highly technical. For frameworks with general graphs, it is very unclear whether one can achieve rerouting that does not decrease the rank and therefore allow an irrelevant vertex argument to go through.

Also, in the lines of [16], an interesting open question is whether we can obtain a deterministic FPT algorithm for MAX RANK (s,t)-PATH for frameworks with matroids not representable in finite fields of small order or in the field of rationals. For example, uniform matroids, and more generally transversal matroids, are representable over a finite field, but the field of representation must be large enough. While the approach of [16] also gives a *randomized* FPT algorithm for frameworks of transversal matroids, our dynamic programming subroutine relies on the efficient computation of representative sets, which requires a linear representation of the input matroid. We stress that this is the only place in the proof of Theorem 1.1 requiring a linear representation of the matroid. However, Theorem 1.2 demonstrates limitations by establishing an unconditional computational lower bound for matroids given by their independence oracles.

REFERENCES

[1] J. Alman and V. V. Williams, *A refined laser method and faster matrix multiplication*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, D. Marx, ed., SIAM, 2021, pp. 522–539, https://doi.org/10.1137/1.9781611976465.32.

[2] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. ACM, 42 (1995), pp. 844–856, https://doi.org/10.1145/210332.210337.

[3] J. Baste, I. Sau, and D. M. Thilikos, *A complexity dichotomy for hitting connected minors on bounded treewidth graphs: The chair and the banner draw the boundary*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2020, pp. 951–970, https://www.doi.org/10.1137/1.9781611975994.57.

[4] A. Björklund, *Determinant sums for undirected Hamiltonicity*, SIAM J. Comput., 43 (2014), pp. 280–299, https://doi.org/10.1137/110839229.

[5] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, *Narrow sieves for parameterized paths and packings*, J. Comput. System Sci., 87 (2017), pp. 119–139, https://doi.org/10.1016/j.jcss.2017.03.003.

[6] A. Björklund, T. Husfeldt, and N. Taslaman, *Shortest cycle through specified elements*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, Y. Rabani, ed., SIAM, 2012, pp. 1747–1753, https://doi.org/10.1137/1.9781611973099.139.

[7] H. Broersma, X. Li, G. Woeginger, and S. Zhang, *Paths and cycles in colored graphs*, Australas. J. Combin., 31 (2005), pp. 299–312, https://ajc.maths.uq.edu.au/pdf/31/ajc_v31_p299.pdf.

[8] G. Călinescu, C. Chekuri, M. Pál, and J. Vondrák, *Maximizing a monotone submodular function subject to a matroid constraint*, SIAM J. Comput., 40 (2011), pp. 1740–1766, https://doi.org/10.1137/080733991.

[9] C. Chekuri and M. Pál, *A recursive greedy algorithm for walks in directed graphs*, in Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 2005, pp. 245–253.

[10] J. Cohen, G. F. Italiano, Y. Manoussakis, N. K. Thang, and H. P. Pham, *Tropical paths in vertex-colored graphs*, J. Comb. Optim., 42 (2021), pp. 476–498, https://doi.org/10.1007/s10878-019-00416-y.

[11] B. Couëtoux, E. Nakache, and Y. Vaxès, *The maximum labeled path problem*, Algorithmica, 78 (2017), pp. 298–318, https://doi.org/10.1007/s00453-016-0155-6.

[12] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, New York, 2015.

[13] R. Diestel, *Graph Theory*, 4th ed., Grad. Texts in Math. 173, Springer, New York, 2012.

[14] E. Eiben, T. Koana, and M. Wahlström, *Determinantal sieving*, in Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2024, pp. 377–423, https://doi.org/10.1137/1.9781611977912.16.

[15] F. V. Fomin, P. A. Golovach, T. Korhonen, D. Lokshtanov, and G. Stamoulis, *Shortest cycles with monotone submodular costs*, in Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA), N. Bansal and V. Nagarajan, eds., SIAM, 2023, pp. 2214–2227, https://doi.org/10.1137/1.9781611977554.ch83.

[16] F. V. Fomin, P. A. Golovach, T. Korhonen, K. Simonov, and G. Stamoulis, *Fixed-parameter tractability of maximum colored path and beyond*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, N. Bansal and V. Nagarajan, eds., SIAM, 2023, pp. 3700–3712, https://doi.org/10.1137/1.9781611977554.ch142.

[17] F. V. Fomin, P. A. Golovach, and D. M. Thilikos, *Modification to planarity is fixed parameter tractable*, in Proceedings of the 36th International Symposium on Theoretical Aspects of Computer Science, LIPIcs. Leibniz Int. Proc. Inform. 126, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 28:1–28:17, https://doi.org/10.4230/LIPIcs.STACS.2019.28.

[18] F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh, *Efficient computation of representative families with applications in parameterized and exact algorithms*, J. ACM, 63 (2016), pp. 29:1–29:60, https://doi.org/10.1145/2886094.

[19] F. V. Fomin, D. Lokshtanov, F. Panolan, S. Saurabh, and M. Zehavi, *Hitting topological minors is FPT*, in STOC 2020: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, ACM, 2020, pp. 1317–1326, https://www.doi.org/10.1145/3357713.3384318.

[20] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos, *Linear kernels for (connected) dominating set on H-minor-free graphs*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, 2012, pp. 82–93, https://www.doi.org/10.1137/1.9781611973099.7.

[21] P. A. Golovach, M. Kaminski, S. Maniatis, and D. M. Thilikos, *The parameterized complexity of graph cyclability*, SIAM J. Discrete Math., 31 (2017), pp. 511–541, https://doi.org/10.1137/141000014.

[22] P. A. Golovach, G. Stamoulis, and D. M. Thilikos, *Model-checking for first-order logic with disjoint paths predicates in proper minor-closed graph classes*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, N. Bansal and V. Nagarajan, eds., SIAM, 2023, pp. 3684–3699, https://doi.org/10.1137/1.9781611977554.ch141.

[23] M. Grohe, K. Kawarabayashi, D. Marx, and P. Wollan, *Finding topological subgraphs is fixed-parameter tractable*, in STOC '11: Proceedings of the 43rd ACM Symposium on Theory of Computing, ACM, 2011, pp. 479–488, https://www.doi.org/10.1145/1993636.1993700.

[24] B. M. P. Jansen, J. J. H. de Kroon, and M. Włodarczyk, *Vertex deletion parameterized by elimination distance and even less*, in STOC 2021: Proceedings of the 53rd Annual ACM Symposium on Theory of Computing, 2021, pp. 1757–1769, https://www.doi.org/10.1145/3406325.3451068.

[25] K. Kawarabayashi, *An improved algorithm for finding cycles through elements*, in Proceedings of the 13th International Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 5035, 2008, pp. 374–384, https://doi.org/10.1007/978-3-540-68891-4_26.

[26] K. Kawarabayashi, *Planarity allowing few error vertices in linear time*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, 2009, pp. 639–648, https://doi.org/10.1109/FOCS.2009.45.

[27] K. Kawarabayashi, S. Kreutzer, and B. Mohar, *Linkless and flat embeddings in 3-space and the unknot problem*, in SoCG '10: Proceedings of the 26th Annual Symposium on Computational Geometry, ACM, 2010, pp. 97–106, https://www.doi.org/10.1145/1810959.1810975.

[28] K. Kawarabayashi, B. Mohar, and B. A. Reed, *A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width*, in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, 2008, pp. 771–780, https://doi.org/10.1109/FOCS.2008.53.

[29] K. Kawarabayashi and B. Reed, *Hadwiger's conjecture is decidable*, in STOC '09: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 2009, pp. 445–454, https://doi.org/10.1145/1536414.1536476.

[30] K. Kawarabayashi and B. Reed, *Odd cycle packing*, in STOC '10: Proceedings of the 42nd ACM Symposium on Theory of Computing, 2010, pp. 695–704, https://www.doi.org/10.1145/1806689.1806785.

[31] J. M. Kleinberg, *Decision algorithms for unsplittable flow and the half-disjoint paths problem*, in Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, ACM, 1998, pp. 530–539.

[32] T. Kloks, *Treewidth, Computations and Approximations*, Lecture Notes in Comput. Sci. 842, Springer, New York, 1994.

[33] Y. Kobayashi and K. Kawarabayashi, *Algorithms for finding an induced cycle in planar graphs and bounded genus graphs*, in Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2009, pp. 1146–1155, https://dl.acm.org/doi/10.5555/1496770.1496894.

[34] T. Korhonen, *A single-exponential time 2-approximation algorithm for treewidth*, in 2021 EEE 62nd Annual Symposium on Foundations of Computer Science (FOCS), 2022, pp. 184–192, https://doi.org/10.1109/FOCS52979.2021.00026.

[35] I. Koutis, *Faster algebraic algorithms for path and packing problems*, in International Colloquium on Automata, Languages and Programming. ICALP 2008. Lecture Notes in Computer Science 5125, Springer-Verlag, 2008, pp. 575–586, https://doi.org/10.1007/978-3-540-70575-8_47.

[36] I. Koutis and R. Williams, *Algebraic fingerprints for faster algorithms*, Comm. ACM, 59 (2015), pp. 98–105, https://doi.org/10.1145/2742544.

[37] D. Lokshtanov, P. Misra, F. Panolan, and S. Saurabh, *Deterministic truncation of linear matroids*, ACM Trans. Algorithms, 14 (2018), pp. 14:1–40:20, https://doi.org/10.1145/3170444.

[38] L. Lovász, *Flats in matroids and geometric graphs*, in Combinatorial Surveys: Proceedings of the 6th British Combinatorial Conference, 1977, pp. 45–86.

[39] L. Lovász, *Graphs and Geometry*, Amer. Math. Soc. Colloq. Publ. 65, American Mathematical Society, Providence, RI, 2019, https://doi.org/10.1090/coll/065.

[40] L. Lovász and M. D. Plummer, *Matching Theory*, North-Holland Math. Stud. 121, North-Holland Publishing, Amsterdam, 1986.

[41] D. Marx, *A parameterized view on matroid optimization problems*, Theoret. Comput. Sci., 410 (2009), pp. 4471–4479, https://doi.org/10.1016/j.tcs.2009.07.027.

[42] D. Marx and I. Schlotter, *Obtaining a planar graph by vertex deletion*, Algorithmica, 62 (2012), pp. 807–822, https://doi.org/10.1007/s00453-010-9484-z.

[43] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, *An analysis of approximations for maximizing submodular set functions*—I, Math. Program., 14 (1978), pp. 265–294, https://doi.org/10.1007/BF01588971.

[44] J. Oxley, *Matroid Theory*, 2nd ed., Oxf. Grad. Texts Math. 21, Oxford University Press, Oxford, 2011, https://doi.org/10.1093/acprof:oso/9780198566946.001.0001.

[45] N. Robertson and P. D. Seymour, *Graph minors* XIII. *The disjoint paths problem*, J. Combin. Theory Ser. B, 63 (1995), pp. 65–110, https://doi.org/10.1006/jctb.1995.1006.

[46] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, J. Combin. Theory Ser. B, 62 (1994), pp. 323–348, https://doi.org/10.1006/jctb.1994.1073.

[47] I. Sau, G. Stamoulis, and D. M. Thilikos, *k-apices of minor-closed graph classes.* II. *Parameterized algorithms*, ACM Trans. Algorithms, 18 (2022), pp. 1–30, https://doi.org/10.1145/3519028.

[48] M. Wahlström, *Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem*, in Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science, LIPIcs. Leibniz Int. Proc. Inform. 20, N. Portier and T. Wilke, eds., Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, pp. 341–352, https://doi.org/10.4230/LIPIcs.STACS.2013.341.