Computing Paths of Large Rank in Planar Frameworks Deterministically

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway

Petr A. Golovach
Department of Informatics, University of Bergen, Norway

Tuukka Korhonen
Department of Informatics, University of Bergen, Norway

Giannos Stamoulis
LIRMM, Université de Montpellier, CNRS, Montpellier, France

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. [SODA 2023] and Eiben et al. [ArXiV 2023] 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 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.

Our main technical contribution is the nontrivial adaptation of the classic irrelevant vertex technique to frameworks to reduce the given instance to one of bounded treewidth. This allows us to employ the toolbox of representative sets to design a dynamic programming procedure solving the problem efficiently on instances of bounded treewidth.

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

Keywords and phrases Planar graph, longest path, linear matroid, irrelevant vertex

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.32

Related Version *Full Version*: https://arxiv.org/abs/2305.01993

Funding The research leading to these results has received funding from the Research Council of Norway via the project BWCA (grant no. 314528). Giannos Stamoulis acknowledges support by the ANR project ESIGMA (ANR-17-CE23-0010) and the French-German Collaboration ANR/DFG Project UTMA (ANR-20-CE92-0027).

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 [37] on representative families of linear matroids and as *matroid graphs* in the book of Lovász and Plummer [38].

The problem of computing maximum matching in frameworks is closely related to the matchoid, the matroid parity, and polymatroid matching problems (see [38] 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 [7, 8, 14, 41].

Fomin et al. in [15] introduced the following MAXIMUM 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 [1].

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 [5, 25, 45]. 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 [6, 15] (see also [9, 10]). 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 $L_1, \ldots, L_t$ and a set I is independent if $|I \cap L_i| \leq 1$ for every label $i \in \{1, \ldots, t\}$. 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 $L_1, \ldots, L_t$.

Randomized FPT algorithms for Maximum Rank (s,t)-Path. The parameterized complexity of MAXIMUM RANK (s,t)-PATH was unknown until very recently. The first FPT algorithm for MAXIMUM RANK (s,t)-PATH was given in [15]. 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 [13], 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 [15] extends the celebrated algebraic technique based on *cancellation of monomials* used by Björklund, Husfeldt, and Taslaman [5] to solve the T-CYCLE problem, while the algorithm of [13] utilizes the toolbox of *(constrained) multilinear detection* [3, 4, 34, 35] combined with *determinantal sieving* [13]. 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 [13] state it, "derandomization appears infeasible" for the algorithms of [15] and [13] for MAXIMUM RANK (s, t)-PATH. Therefore, the next challenge is to obtain *derandomized* FPT algorithms for this problem.

Our results. Our main result establishes the first *deterministic* FPT algorithm for MAXIMUM RANK (s, t)-PATH on frameworks of planar graphs and matroids representable over finite fields or over the field of rationals. We use $|G|$ to denote the number of vertices of a graph G and $\|M\|$ to denote the bit-length of the representation matrix of a linear matroid M.

▶ **Theorem 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.*

Note that the randomized FPT algorithms of [15] and [13] work for matroids representable over finite fields or fields of characteristic two. The algorithm of Theorem 1, apart from being the first deterministic algorithm for MAXIMUM 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 $\mathbb{Q}$.

Our techniques. To design the deterministic FPT algorithm of Theorem 1, we follow a different proof strategy than that of [15] and [13]. Our approach is based on the *win/win* arguments of the celebrated *irrelevant vertex technique* of Robertson and Seymour [42]. 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 can not be further applied, the obtained reduced instance is equivalent to the original one and also "simpler". Therefore, one 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, [2, 17, 19, 20, 22–30, 32, 40, 44] (see also [11, Section 7.8]). The standard mesure 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 [42] 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 MAXIMUM 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. Planarity of the input graph allows to find such an area using the grid-like structure of *walls*. An overview of this approach is provided in Subsection 1.1. This application of the irrelevant vertex technique for frameworks is novel and illustrates an interesting interplay between combinatorial structures and algebraic properties, that may be of independent interest.

The dynamic programming on graphs of bounded treewidth is pretty standard (see, e.g., [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, 36]) instead of all possible independent sets. Both randomized [18] and deterministic [36] constructions of representative sets require a linear representation of M. This is the reason why Theorem 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 requiring a representation of M. It is an interesting open question, whether Maximum Rank (s,t)-Path is FPT when parameterized by k and the treewidth if the input matroid is given by its independence oracle.

1.1 Overview of the proof of Theorem 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 Maximum 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 [43] 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.

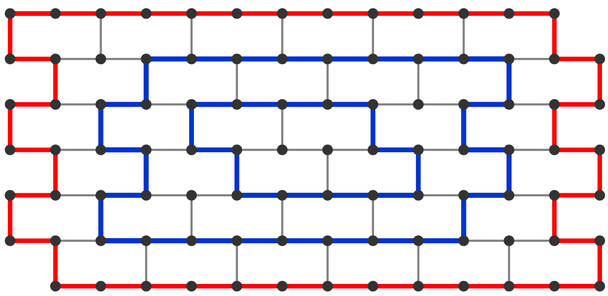

Figure 1 A 7-wall and its layers.

Note that an h-wall has $\lfloor h/2 \rfloor$ nested cycles, called *layers*, that are shown in Figure 1 in red and blue. 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. The vertices of the initial h-wall, i.e., before replacing edges by paths, are called the *branch vertices* of the subdivided h-wall. 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 $\mathsf{compass}(W)$. 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.

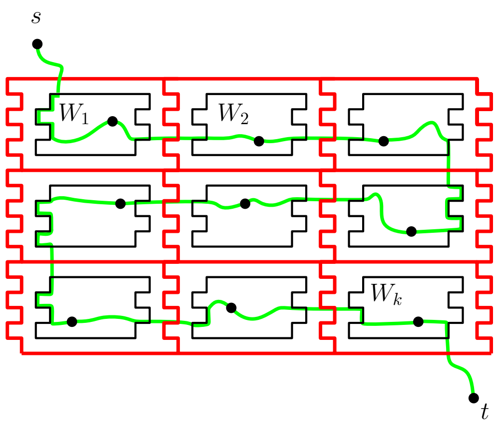

Figure 2 An (s,t)-path for walls of big rank.

Suppose that there is a packing of k subwalls $W_1,\dots,W_k$ in W separated by paths in W as it is 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 it is 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 $i \in \{0,\dots,k\}$.
- 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 it is 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 y are unique vertices of these paths in the inner insulation layer, and (ii) $u_1,\dots,u_\ell \in V(P_1)\cup V(P_2)$. The proof that $\mathcal{O}(k^2)$ insulation layers are sufficient for rerouting P is non-trivial. 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 $i \in \{1,\dots,k\}$, 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 $i \in \{1,\dots,k\}$. Therefore, one can select, for every W_i, a vertex $v_i \in I_i$ and this choice can be made so that $r(\{v_1,\dots,v_k\}) = r(\mathsf{compass}(W'))$. 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 $v_1,\dots,v_k$, and (iii) Q avoids W_0. We have that $P' = P_1QP_2$ is an (s,t)-path that goes through $u_1,\dots,u_\ell$ and $v_1,\dots,v_k$. Note that, replacing the vertices of X that are spanned by $V(\mathsf{compass}(W'))$ by the vertices $\{v_1,\dots,v_k\}$, we obtain the set $X' = \{u_1,\dots,u_\ell,v_1,\dots,v_k\}$ and $r(X') = r(X)$, and the latter holds since $r(\{v_1,\dots,v_k\}) = r(\mathsf{compass}(W'))$. Therefore $r(P') \geq r(X) \geq k$. Since Q avoids W_0, P' has the same property.

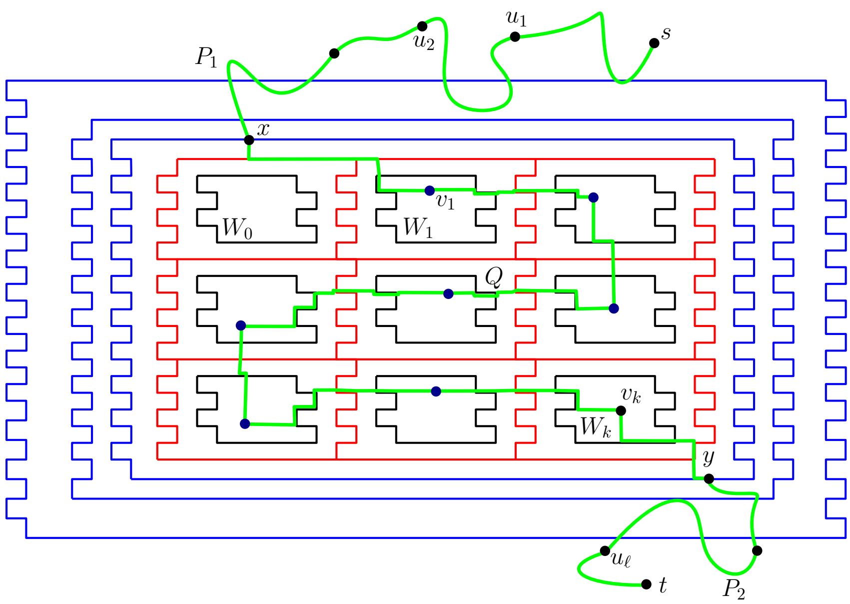

Figure 3 Rerouting an (s,t)-path.

Finally, we note that the algorithm of Kawarabayashi [25] for T-CYCLE works for general graphs. The statement of Theorem 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 towards pushing the irrelevant vertex technique from frameworks with planar graphs to frameworks with general graphs.

Organization of the paper. In Section 2 we show how to reduce to instances of bounded treewidth using the irrelevant vertex technique. Results whose proofs are omitted in this extended abstract are marked with a star ($\star$) and their proofs can be found in the full version [16]. We also refer to the full version for formal definitions of the aforementioned notions. We conclude in Section 3 with open questions and possible future research directions.

2 Rerouting paths and cycles

In this section, our goal is to prove Theorem 1 that we restate here.

▶ **Theorem 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 consists of two parts. In the first part, we use the irrelevant vertex technique in order to design an algorithm that removes vertices form the input graph as long as its treewidth is big enough. In order to do this, we show a combinatorial result 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 (Lemma 4). Then, we give an algorithm (Lemma 6) 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. The dynamic programming algorithm that solves the problem in graphs of bounded treewidth is included in the full version of the paper.

2.1 Rerouting paths and cycles

In this subsection, we aim to prove the main combinatorial result (Lemma 4) 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 4, we state the following result (Lemma 3) that will be an important tool for the proof of Lemma 4. The proof of Lemma 3 is inspired by the proof of [25, Lemma 1]. An *in-peg* of the perimeter of a wall W is a vertex on the perimeter of W that has degree three in W.

▶ **Lemma 3** ($\star$). *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 $E = \{e_1, \ldots, e_k, e_{k+1}, e_{k+2}\}$ be a set of $k + 2$ edges of G, where, for every $i \in \{1, \ldots, k\}$, $e_i = \{v_i, u_i\}$, $e_{k+1} = \{v_{k+1}, s\}$, $e_{k+2} = \{v_{k+2}, t\}$, and let X be the set $\{v_{k+1}, v_{k+2}\} \cup \bigcup_{i \in \{1,\ldots,k\}} \{v_i, u_i\}$. If every $v \in X$ is an in-peg of $\mathsf{perim}(W)$, then there is an (s, t)-path in G that contains the edges $e_1, \ldots, e_{k+2}$ 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.*

We now prove the following result.

▶ **Lemma 4.** *There is a function $h : \mathbb{N} \to \mathbb{N}$ such that if $k, z \in \mathbb{N}$, 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 $V(\mathsf{inn}(W^{(h(k))}))$, 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 $i \in \{1, \ldots, k + 2\}$, 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 $i \in \{2, \ldots, k + 2\}$, 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 4 for an example).

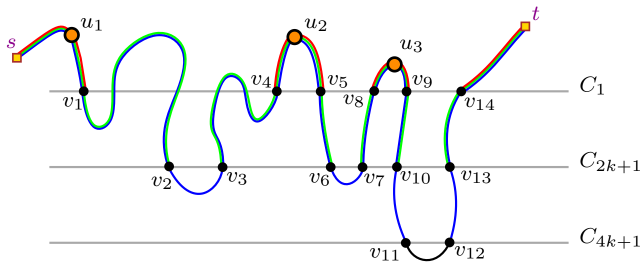

Figure 4 An example of an (s,t)-path P containing an independent set $S = \{u_1, u_2, u_3\}$. In this example, $\mathcal{F}_1$ is the collection of the four red paths (the ones with endpoints (s, v_1), (v_4, v_5), (v_8, v_9), and (v_{14}, t)), $\mathcal{F}_2$ is the collection of the four green paths (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).

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 $i \in \{1, \ldots, k+2\}$, we assume that $\mathcal{F}_i = \{F_{i,1}, \ldots, F_{i,|\mathcal{F}_i|}\}$, where the ordering is given by traversing P from s to t. For every $i \in \{1, \ldots, k+2\}$, we set $\mathcal{Q}_i = \{Q_{i,1}, \ldots, Q_{i,|\mathcal{F}_i|-1}\}$, 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 $i \in \{1, \ldots, k+2\}$, P is the concatenation of the paths $F_{i,1}, Q_{i,1}, F_{i,2}, \ldots, Q_{i,|\mathcal{F}_i|-1}, F_{i,|\mathcal{F}_i|}$. In Figure 4, $\mathcal{Q}_1 = \{Q_{1,1}, Q_{1,2}, Q_{1,3}\}$, 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 $i \in \{1, \ldots, k+1\}$, $|\mathcal{F}_{i+1}|$ 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 $i \in \{1, \ldots, k+1\}$, $|\mathcal{F}_i| \geq |\mathcal{F}_{i+1}|$, there is an $i_0 \in \{1, \ldots, k+1\}$ 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}.

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 4, $|\mathcal{F}_1| = |\mathcal{F}_2|$ 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 5).

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 three in $W^{(2k\cdot i_0)}$. Also, we set $\mathcal{K}$ to be the graph $G' \setminus V(\mathsf{inn}(W^{(2k\cdot i_0)}))$. The next claim states that there also exist ℓ disjoint paths from Y to B in $\mathcal{K}$. We omit the proof and we refer the reader to the full version [16].

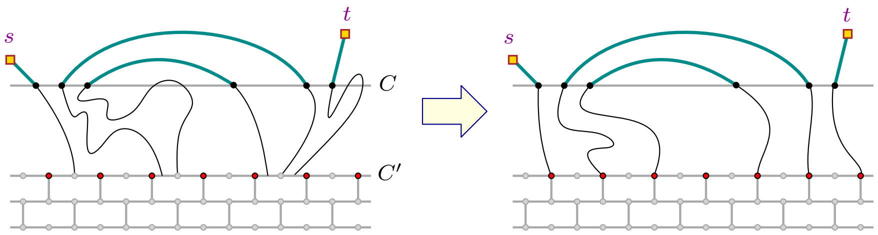

Figure 5 A visualization of the statement of Claim 5. In both figures, the edges $\{v_F, u_F\}$ are depicted in blue, the black vertices correspond to the set Y and the red vertices correspond to the set B. In the left figure, we illustrate $|Y|$ disjoint paths from Y to C', while in the right figure, we illustrate $|Y|$ disjoint paths from Y to B.

▷ Claim 5 ($\star$). 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 $\mathcal{K}$.

Following Claim 5, let $X \subseteq B$, let a bijection $\rho : Y \to X$, and let 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, 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 of $\mathsf{perim}(W^{(2k\cdot i_0+k+1)})$ whose endpoints, say x and y, are branch vertices of W. Finally, let an (x,y)-path $R_{x,y}$ in $\mathsf{compass}(W^{(2k\cdot i_0+k+1)})$ whose 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 that $(P^\star \setminus V(\mathsf{compass}(W^{(2k\cdot i_0+k+1)}))) \cup R_{x,y}$ is the (s,t)-path claimed in the statement of the lemma. ◀

We stress that, while Lemma 4 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.

2.2 Equivalent instances of small treewidth

In this subsection, we design 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 6.** *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* Maximum 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 6, 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 let q be a non-negative 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 $i \in \{0, \ldots, r-1\}$, 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 $i, j \in \{0, \ldots, r-1\}$, 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 7.** *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 8** ($\star$)**.** *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, $f(k, z, r, q) = \mathcal{O}(r^{k/2} \cdot z \cdot q)$.*

We are now ready to prove Lemma 6.

Proof of Lemma 6. We set $b = h(k)$, $x = k+1$, $z = (k+1) \cdot b$, $q = f(k-1, z, x, 3)$, $r = \lceil \sqrt{k} \rceil \cdot (q+1) + 3$, and $g(k) = 36(r+1)$. 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 $g(k) = 36(r+1)$, due to [21, Lemma 4.2], G has a $(4r+1)$-wall. We then consider an r-wall W of G such that $s, t \notin V(\mathsf{compass}(W))$ 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 7 and we can find it in $\mathcal{O}(n)$ time. For every $i \in \{1, \ldots, k\}$, we set $K_i := V(\mathsf{compass}(\tilde{W}_i))$. Then, compute the rank of K_i, for each $i \in \{1, \ldots, k\}$. 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 $i \in \{1, \ldots, k\}$, $|S \cap K_i| = 1$. 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 $i \in \{1, \ldots, k\}$ such that the rank of K_i is at most $k-1$. Since the corresponding wall $\tilde{W}_i$ is of height at least $q = f(k-1, z, x, 3)$, by Lemma 8, 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', so that for every $i \in \{0, \ldots, k\}$, $\rho(W_i) = \rho(W')$. Let v 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 Maximum 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 a 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 $i \in \{1, \ldots, k+1\}$ 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. By Lemma 4, there is a path $\tilde{P}$ such that $S_{\text{out}} \subseteq V(\tilde{P})$ 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 $i \in \{0, \ldots, k\}$, since otherwise $S_{\text{in}} = \emptyset$ and the claim holds trivially. For every $i \in \{0, \ldots, k\}$, 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 $i \in \{0, \ldots, k\}$, we can choose a set $S' = \{v_1, \ldots, v_k\}$ where v_i is a vertex in S_i for $i \in \{1, \ldots, k\}$ 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). ◀

2.3 Proof of Theorem 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 6. Keep in mind that $g(k) = 2^{\mathcal{O}(k \log k)}$. We describe an algorithm $\mathcal{A}$ that solves Maximum Rank (s,t)-Path.

Our algorithm $\mathcal{A}$ first calls the single-exponential time 2-approximation algorithm for treewidth of Korhonen [33] 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 can solve the problem using our dynamic programming algorithm which runs in time $2^{q^{\mathcal{O}(1)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)} = 2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$. In the second possible output (i.e., where G has treewidth at least q), we apply the algorithm of Lemma 6 and, in time $2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$, we either report a positive answer, 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 the problem. 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 $2^{2^{\mathcal{O}(k \log k)}} \cdot (|G| + \|M\|)^{\mathcal{O}(1)}$.

3 Conclusion

In this paper, we provide a deterministic FPT algorithm for Maximum 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. Let us conclude by discussing some open research directions.

Since the algorithm of [15] for Maximum Rank (s,t)-Path runs in time $2^{\mathcal{O}(k^2 \log(k+q))} n^{\mathcal{O}(1)}$, 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. 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 [15] 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 [15], an interesting open question is whether we can obtain a deterministic FPT algorithm for Maximum 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 [15] 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 requiring a linear representation of the matroid. Another interesting open question, is whether Maximum Rank (s,t)-Path is FPT when parameterized by k and the treewidth if the input matroid is given by its independence oracle.

References

1. Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
2. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *Proc. of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 951–970, 2020. doi:10.1137/1.9781611975994.57.
3. Andreas Björklund. Determinant sums for undirected hamiltonicity. *SIAM J. Comput.*, 43(1):280–299, 2014. doi:10.1137/110839229.
4. Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *J. Comput. Syst. Sci.*, 87:119–139, 2017. doi:10.1016/j.jcss.2017.03.003.
5. Andreas Björklund, Thore Husfeldt, and Nina Taslaman. Shortest cycle through specified elements. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1747–1753. SIAM, 2012. doi:10.1137/1.9781611973099.139.
6. Hajo Broersma, Xueliang Li, Gerhard J Woeginger, and Shenggui Zhang. Paths and cycles in colored graphs. *Australas. J Comb.*, 31:299–312, 2005.
7. Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. doi:10.1137/080733991.
8. Chandra Chekuri and Martin Pál. A recursive greedy algorithm for walks in directed graphs. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 245–253. IEEE Computer Society, 2005.
9. Johanne Cohen, Giuseppe F. Italiano, Yannis Manoussakis, Nguyen Kim Thang, and Hong Phong Pham. Tropical paths in vertex-colored graphs. *J. Comb. Optim.*, 42(3):476–498, 2021. doi:10.1007/s10878-019-00416-y.
10. Basile Couëtoux, Elie Nakache, and Yann Vaxès. The maximum labeled path problem. *Algorithmica*, 78(1):298–318, 2017. doi:10.1007/s00453-016-0155-6.
11. Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
12. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
13. Eduard Eiben, Tomohiro Koana, and Magnus Wahlström. Determinantal sieving. *CoRR*, abs/2304.02091, 2023. doi:10.48550/arXiv.2304.02091.
14. Fedor V. Fomin, Petr A. Golovach, Tuukka Korhonen, Daniel Lokshtanov, and Giannos Stamoulis. Shortest cycles with monotone submodular costs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2214–2227. SIAM, 2023. doi:10.1137/1.9781611977554.ch83.
15. Fedor V. Fomin, Petr A. Golovach, Tuukka Korhonen, Kirill Simonov, and Giannos Stamoulis. Fixed-parameter tractability of maximum colored path and beyond. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3700–3712. SIAM, 2023. doi:10.1137/1.9781611977554.ch142.
16. Fedor V. Fomin, Petr A. Golovach, Tuukka Korhonen, and Giannos Stamoulis. Computing paths of large rank in planar frameworks deterministically. *CoRR*, abs/2305.01993, 2023. doi:10.48550/arXiv.2305.01993.
17. Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Modification to Planarity is Fixed Parameter Tractable. In *Proc. of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2019.28.

18 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.

19 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Hitting topological minors is FPT. In *Proc. of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1317–1326. ACM, 2020. doi:10.1145/3357713.3384318.

20 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on H-minor-free graphs. In *Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 82–93, 2012. doi:10.1137/1.9781611973099.7.

21 Petr A. Golovach, Marcin Kaminski, Spyridon Maniatis, and Dimitrios M. Thilikos. The parameterized complexity of graph cyclability. *SIAM J. Discret. Math.*, 31(1):511–541, 2017.

22 Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Model-checking for first-order logic with disjoint paths predicates in proper minor-closed graph classes. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3684–3699. SIAM, 2023. doi:10.1137/1.9781611977554.ch141.

23 Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In *Proc. of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 479–488. ACM, 2011. doi:10.1145/1993636.1993700.

24 Bart M. P. Jansen, Jari J. H. de Kroon, and Michal Włodarczyk. Vertex deletion parameterized by elimination distance and even less. In *Proc. of the 53rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1757–1769, 2021. doi:10.1145/3406325.3451068.

25 Ken-ichi Kawarabayashi. An improved algorithm for finding cycles through elements. In *13th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 5035 of *Lecture Notes in Computer Science*, pages 374–384. Springer, 2008. doi:10.1007/978-3-540-68891-4_26.

26 Ken-ichi Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proc. of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–648, 2009. doi:10.1109/FOCS.2009.45.

27 Ken-ichi Kawarabayashi, Stephan Kreutzer, and Bojan Mohar. Linkless and flat embeddings in 3-space and the unknot problem. In *Proc. of the 2010 Annual Symposium on Computational Geometry (SoCG)*, pages 97–106. ACM, 2010. doi:10.1145/1810959.1810975.

28 Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *Proc. of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 771–780, 2008. doi:10.1109/FOCS.2008.53.

29 Ken-ichi Kawarabayashi and Bruce A. Reed. Hadwiger's conjecture is decidable. In *Proc. of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 445–454, 2009. doi:10.1145/1536414.1536476.

30 Ken-ichi Kawarabayashi and Bruce A. Reed. Odd cycle packing. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 695–704, 2010. doi:10.1145/1806689.1806785.

31 Jon M. Kleinberg. Decision algorithms for unsplittable flow and the half-disjoint paths problem. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 530–539. ACM, 1998.

32 Yusuke Kobayashi and Ken-ichi Kawarabayashi. Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In *Proc. of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1146–1155. ACM-SIAM, 2009. URL: https://dl.acm.org/doi/10.5555/1496770.1496894.

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

34 Ioannis Koutis. Faster algebraic algorithms for path and packing problems. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming – Volume Part I*, ICALP '08, pages 575–586, Berlin, Heidelberg, 2008. Springer-Verlag. doi:10.1007/978-3-540-70575-8_47.

35 Ioannis Koutis and Ryan Williams. Algebraic fingerprints for faster algorithms. *Commun. ACM*, 59(1):98–105, December 2015. doi:10.1145/2742544.

36 Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. Deterministic truncation of linear matroids. *ACM Trans. Algorithms*, 14(2):14:1–14:20, 2018. doi:10.1145/3170444.

37 L. Lovász. Flats in matroids and geometric graphs. In *Combinatorial surveys (Proc. Sixth British Combinatorial Conf., Royal Holloway Coll., Egham, 1977)*, pages 45–86, 1977.

38 L. Lovász and M. D. Plummer. *Matching theory*, volume 121 of *North-Holland Mathematics Studies*. North-Holland Publishing Co., Amsterdam; North-Holland Publishing Co., Amsterdam, 1986. Annals of Discrete Mathematics, 29.

39 László Lovász. *Graphs and geometry*, volume 65 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI, 2019. doi:10.1090/coll/065.

40 Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012. doi:10.1007/s00453-010-9484-z.

41 George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions – I. *Math. Program.*, 14(1):265–294, 1978. doi:10.1007/BF01588971.

42 Neil Robertson and Paul D. Seymour. Graph minors XIII. The disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006.

43 Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Comb. Theory, Ser. B*, 62(2):323–348, 1994. doi:10.1006/jctb.1994.1073.

44 Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k-apices of minor-closed graph classes. II. Parameterized algorithms. *ACM Transactions on Algorithms*, 18(3), 2022. doi:10.1145/3519028.

45 Magnus Wahlström. Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 – March 2, 2013, Kiel, Germany*, volume 20 of *LIPIcs*, pages 341–352. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.STACS.2013.341.