



Revisiting Directed Disjoint Paths on tournaments (and relatives)

Guilherme C. M. Gomes ✉ 

LIRMM, Université de Montpellier, CNRS, Montpellier, France
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Raul Lopes ✉ 

LIRMM, Université de Montpellier, CNRS, Montpellier, France
Hamburg University of Technology, Institute for Algorithms and Complexity, Hamburg, Germany

Ignasi Sau ✉ 

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Abstract

In the DIRECTED DISJOINT PATHS problem (k -DDP), we are given a digraph and k pairs of terminals, and the goal is to find k pairwise vertex-disjoint paths connecting each pair of terminals. Bang-Jensen and Thomassen [SIAM J. Discrete Math. 1992] claimed that k -DDP is NP-complete on tournaments, and this result triggered a very active line of research about the complexity of the problem on tournaments and natural superclasses. We identify a flaw in their proof, which has been acknowledged by the authors, and provide a new NP-completeness proof. From an algorithmic point of view, Fomin and Pilipczuk [J. Comb. Theory B 2019] provided an FPT algorithm for the edge-disjoint version of the problem on semicomplete digraphs, and showed that their technique cannot work for the vertex-disjoint version. We overcome this obstacle by showing that the version of k -DDP where we allow congestion c on the vertices is FPT on semicomplete digraphs provided that c is greater than $k/2$. This is based on a quite elaborate irrelevant vertex argument inspired by the edge-disjoint version, and we show that our choice of c is best possible for this technique, with a counterexample with no irrelevant vertices when $c \leq k/2$. We also prove that k -DDP on digraphs that can be partitioned into h semicomplete digraphs is W[1]-hard parameterized by $k + h$, which shows that the XP algorithm presented by Chudnovsky, Scott, and Seymour [J. Comb. Theory B 2019] is essentially optimal.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics → Graph theory → Graph algorithms.

Keywords and phrases directed graphs, tournaments, semicomplete digraphs, directed disjoint paths, congestion, parameterized complexity, directed pathwidth.

Related Version A conference version of this article will appear in the proceedings of *ICALP 2025*. We would like to thank the reviewers for their helpful remarks.

Funding *Guilherme C. M. Gomes*: Funded by the European Union, project PACKENUM, grant number 101109317. Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Raul Lopes: French project ELIT (ANR-20-CE48-0008-01) and HIDSS-0002 DASHH (Data Science in Hamburg - Helmholtz Graduate School for the Structure of Matter).

Ignasi Sau: French project ELIT (ANR-20-CE48-0008-01).

1 Introduction

The DISJOINT PATHS problem is one of the most well-studied classical NP-complete graph problems [29]. It consists in, given an undirected graph G and k requests, which are pairs of vertices $(s_1, t_1), \dots, (s_k, t_k)$ known as *terminals*, deciding whether G contains k

pairwise vertex-disjoint paths connecting each s_i to t_i , for each $i \in [k]$. As a crucial ingredient of their Graph Minors project, Robertson and Seymour [40] proved that DISJOINT PATHS is *fixed-parameter tractable* (FPT) parameterized by k , that is, it can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ on n -vertex graphs for some computable function f . In this article we focus on its directed counterpart, defined analogously for an input digraph D and called DIRECTED DISJOINT PATHS, which is known to be much harder from a computational point of view: it is already NP-complete for a fixed number $k = 2$ of terminals [25]. A number of approaches have been proposed to cope with this intractability, ranging from approximation algorithms [12, 43], heuristics [7, 38], parameterized algorithms [23, 41], restricting the input digraph D [2, 16, 25–27], or relaxing the problem by allowing congestion on vertices [1, 10, 20, 31, 33, 34, 37]. In this work we consider and combine the latter three approaches, which we proceed to discuss.

Let us start by restricting the graph class to which the input graph of DIRECTED DISJOINT PATHS belongs. Two relevant classes of digraphs are typically considered when one seeks to improve the tractability of a problem: directed acyclic graphs (DAGs) and *tournaments* (that is, digraphs that can be obtained from a complete graph by orienting each edge). For the former, the (parameterized) complexity of DIRECTED DISJOINT PATHS is well understood: the problem is NP-complete and solvable in time $n^{\mathcal{O}(k)}$ (hence, in the class XP) [25], and W[1]-hard [41] (even to approximate within a constant factor [43]), thus unlikely to be FPT. For the latter, the landscape is more murky, and the goal of this article is to contribute to understanding it a bit better.

Bang-Jensen and Thomassen [2] *claimed* that DIRECTED DISJOINT PATHS is NP-complete on tournaments when k is part of the input, and this result triggered a long line of research. In the same paper, Bang-Jensen and Thomassen [2] showed that the problem can be solved in polynomial time for $k = 2$. In fact, their algorithm works on the larger class of *semicomplete digraphs*, where each pair of distinct vertices has at least one arc between them, instead of exactly one as in tournaments. Chudnovsky, Scott, and Seymour [16] showed that the problem can be solved in polynomial time for every fixed k on semicomplete digraphs, by providing an XP algorithm. Fradkin and Seymour [27] proved that the edge-disjoint version of the problem, for which all the results discussed above also apply, is also polynomial-time solvable on semicomplete digraphs for fixed k . It is worth mentioning that Chudnovsky, Fradkin, Kim, Scott, and Seymour [13, 15, 16, 26, 27, 35] built a containment theory on semicomplete digraphs for the study of minor-related problems, such as DIRECTED DISJOINT PATHS (see also the work of Barbero, Paul, and Pilipczuk [5]). One of the key notions in this theory is a width measure of digraphs called *directed pathwidth* – a generalization of undirected pathwidth (see Section 3 for the definition) – that also plays a role in the current article.

Back to the DIRECTED DISJOINT PATHS problem, the XP algorithm on tournaments [16] has been generalized by Chudnovsky, Scott, and Seymour [14] to the class \mathcal{C}_h of digraphs whose vertex set can be partitioned into a bounded number h of semicomplete digraphs (thus, semicomplete digraphs correspond to the class \mathcal{C}_1). More precisely, the running time of their algorithm is $n^{\mathcal{O}((hk)^5)}$. The authors asked whether the result can be further generalized to the class \mathcal{A}_h of digraphs whose underlying graph has independence number at most h , and this is still open. For the edge-disjoint version of the problem, an affirmative answer was given by Fradkin and Seymour [27].

Concerning the fixed-parameter tractability of the problem, it is open whether DIRECTED DISJOINT PATHS on tournaments (or semicomplete digraphs) is FPT parameterized by k . Interestingly, the edge-disjoint version of the problem was shown to be FPT on semicomplete digraphs by Fomin and Pilipczuk [23], by solving a more general problem called ROOTED

IMMERSION. In a nutshell (more details are given below when discussing our techniques in Section 2), there is a win/win approach based on the directed pathwidth of the input digraph. If the pathwidth is small (as a function of k), then a dynamic programming algorithm is used to solve the problem. Otherwise, it is shown that the digraph must contain a large obstruction to directed pathwidth called a *triple* [26], which is used to find an *irrelevant vertex* for the problem, that is, a vertex whose removal does not affect the existence of a solution.

Note that the above win/win approach needs, as a first step, to compute the directed pathwidth of the input digraph in FPT-time. While this problem is open on general digraphs, an FPT algorithm on semicomplete digraphs is given in the same article [23]. This FPT algorithm to compute directed pathwidth was generalized by Kitsunai, Kobayashi, and Tamaki [36] to another superclass of semicomplete digraphs called *h -semicomplete digraphs* and denoted by \mathcal{S}_h for some fixed integer $h \geq 0$. Digraphs in this class satisfy the property that each vertex has at most h non-neighbors (in any direction). If we denote by \mathcal{T} and \mathcal{S} the classes of tournaments and semicomplete digraphs, respectively, it holds that $\mathcal{S} = \mathcal{S}_0$, and for every $h \geq 0$ (see the discussion in the beginning of Subsection 3.2),

$$\mathcal{T} \subseteq \mathcal{S} \subseteq \mathcal{S}_h \subseteq \mathcal{C}_{h+1} \subseteq \mathcal{A}_{h+1}. \quad (1)$$

Another transversal strategy to try to overcome the inherent hardness of DIRECTED DISJOINT PATHS is to relax the problem by allowing every vertex of D to be used by at most c of the k paths of the solution, for some integer $c \geq 1$ that is also part of the input and is called the *congestion*; we call the corresponding problem DIRECTED DISJOINT PATHS WITH CONGESTION. Note that if $c \geq k$ the problem is trivial, so we may assume that $c \leq k - 1$. It is open whether the case $k = 3$ and $c = 2$ (which is the first non-trivial one with congestion greater than one) can be solved in polynomial time on general digraphs. There are, however, some positive results. For instance, Edwards, Muzi, and Wollan [20] showed that the problem for $c = 2$ can be solved in polynomial time if the input graph is sufficiently connected as a function of k , which is not the case for the truly disjoint version [42]. See [10] for recent improvements of this result. A popular variant of the congested problem is an *asymmetric* version, where the goal is to either find a congested solution or to provide a no-answer for the disjoint version. This problem has been proved to be XP parameterized by k on general digraphs for some small values of c [31, 33, 34], usually exploiting the celebrated Directed Grid Theorem [9, 32, 34]. On the other hand, other articles [1, 37] study the parameterized complexity of DIRECTED DISJOINT PATHS WITH CONGESTION by considering parameters stronger than k .

Finally, it is worth mentioning that Cavallaro, Kawarabayashi, and Kreutzer [11] recently proved that the edge-disjoint version of DIRECTED DISJOINT PATHS is FPT on Eulerian digraphs parameterized by k ; this is one of the rare examples where (some variant of) the problem is FPT. We refer to the book of Bang-Jensen and Gutin [4] for a thorough introduction to algorithms on digraphs, in particular on tournaments and related superclasses.

In the sequel, for notational conciseness we may use the abbreviations k -DDP and (k, c) -DDP to refer to the DIRECTED DISJOINT PATHS and DIRECTED DISJOINT PATHS WITH CONGESTION problems, respectively. We permit ourselves to slightly abuse notation by including the integers k and c in the abbreviated names of the problems, even if they are part of the corresponding inputs.

Our contributions. As mentioned above, the NP-completeness proof of Bang-Jensen and Thomassen [2] of k -DDP on tournaments triggered intensive research in this area [14, 16, 23].

Unfortunately, we realized that their proof has a flaw that does not seem to be easily fixable, as acknowledged by the authors [3]; see [Appendix A](#) for an explanation of this flaw. Our first contribution is to provide a new (correct) proof of this result.

► **Theorem 1.** *DIRECTED DISJOINT PATHS on tournaments is NP-complete.*

As mentioned above, it is open whether k -DDP on tournaments is FPT parameterized by k . Recall that the win/win approach of Fomin and Pilipczuk [23] for the edge-disjoint version has two main ingredients (other than computing the directed pathwidth): a dynamic programming algorithm and an irrelevant vertex argument. While the former is claimed to exist for the vertex-disjoint version [23], the latter one is doomed to fail: Fomin and Pilipczuk [23] provide a counterexample even for $k = 2$ consisting of a family of tournaments containing arbitrarily large triples (that are the structures where irrelevant vertices are found), but in which each vertex is relevant. Our next contribution is to prove that this obstacle disappears if we allow for a large congestion.

► **Theorem 2.** *(k, c) -DDP on semicomplete digraphs is FPT parameterized by k restricted to instances satisfying $c > k/2$.*

Note that since we can assume that $c \leq k-1$, the result of [Theorem 2](#) covers roughly “half” of the range of values of the congestion c . It is natural to ask whether the problem remains NP-complete for this range of values of c , that is, when the congestion is lower-bounded by a linear function of k . This question is still open, but we provide the following hardness result, where the congestion c is *almost linear* in k (as ε approaches one).

► **Theorem 3.** *(k, c) -DDP remains NP-complete on tournaments even restricted to instances satisfying $c = k^\varepsilon$, for every $\varepsilon \in [0, 1)$.*

As discussed before, Chudnovsky, Scott, and Seymour [14] showed that DIRECTED DISJOINT PATHS on the class \mathcal{C}_h can be solved in time $n^{\mathcal{O}((hk)^5)}$. Our next result is to show that this algorithm is somehow optimal, in the sense that it is unlikely to get rid of both parameters in the exponent of n , even restricted to digraphs of bounded directed pathwidth.

► **Theorem 4.** *The k -DDP problem on \mathcal{C}_h is W[1]-hard when parameterized by $k + h$, even if restricted to input digraphs of directed pathwidth two.*

Moreover, [Theorem 4](#) can be generalized to show hardness for (k, c) -DDP when $c > k/2$ (cf. [Theorem 19](#), which, alongside [Equation 1](#), implies that the win/win strategy cannot be extended beyond h -semicomplete graphs). We summarize our main contributions in [Table 2](#).

Organization. In [Section 2](#) we give an overview of the techniques used to obtain our results. In [Section 3](#) we provide preliminaries about general digraphs, tournaments, related classes, directed pathwidth, and parameterized complexity. In [Section 4](#) (resp. [Section 5](#)) we provide our negative (resp. positive) results. We conclude the article in [Section 6](#) with some directions for further research.

2 Overview of our techniques

The reduction that we use to prove [Theorem 1](#) is novel and versatile enough so that we can build upon it and modify it appropriately to prove the NP-completeness of several variants of the problem. Intuitively, in the proof of [Theorem 1](#) we reduce from a variant of 3-SAT

where each variable has a bounded number of occurrences (namely, exactly three positive and one negative). This allows us to build an instance of k -DDP where a variable's truth value is determined by two requests (cf. Figure 3), while the satisfaction of the clauses is encoded using one additional request per clause; interestingly, the paths that fulfill the requests have length at most five. By extending this construction using a single long path, named the *critical path* (cf. the black path in Figure 5), which is the unique way to fulfill several additional requests, we show how to prove NP-hardness for (k, c) -DDP for digraphs in \mathcal{C}_2 even if $c = \varepsilon k$ for $\varepsilon \in [0, 1)$ (cf. Theorem 11). With this approach, however, we are unable to prove hardness for (k, c) -DDP instances where $c = \varepsilon k$ and D is a tournament, only doing so for the weaker relation $c = k^\varepsilon$ (cf. Theorem 3).

For the FPT algorithm given in Theorem 2, the challenge is that now we deal with vertex-disjoint paths, instead of edge-disjoint paths as in the original work by Fomin and Pilipczuk [23]. Much like theirs, our proof is a win/win approach that makes extensive use of k -triples (cf. Definition 5), an obstacle for directed pathwidth on tournaments introduced by Fradkin and Seymour [26], and has two steps: (i) show that, in a minimum solution, almost all vertices (in their case, arcs) of a sufficiently large (i.e., with more than $f(k)$ vertices) triple can be used by other paths, and (ii) identify a vertex in the triple that can be safely removed from the instance. In the edge-disjoint setting, it was extensively used that, in a large enough triple, vertices had large in- or out-degree; consequently, finding available arcs to either shortcut (step i) or reroute (step ii) a solution was relatively simple. In the vertex-disjoint case, this does not happen; in fact, Fomin and Pilipczuk [23] present an infinite family of tournaments that have no irrelevant vertex for k -DDP. As we show in Subsection 3.3, their counterexamples are also valid when $c \leq k/2$ but, when $c > k/2$, we are able to easily find alternative paths, freeing up several vertices and thus making them irrelevant. In particular, we are able to implement step i using, in particular, the pigeonhole principle: any two fully congested vertices have at least one path in common. This is not enough by itself, and we must carefully construct two shortcuts simultaneously, instead of only one, to show that vertices occupied by exactly c paths only amount to $\mathcal{O}\left(\binom{k}{c}\right)$ (see Lemma 25 elements of the triple). Step ii also offers additional challenges; in particular, when dealing with the exterior neighborhood of the triple, we must find large matchings entering and leaving the triple to properly reroute the paths using it. As such, instead of the polynomial on k used by Fomin and Pilipczuk, we now require that the triple has size of the order of $2^{\mathcal{O}(k \log k)}$.

Given the success of the win/win algorithms based on directed pathwidth, it is natural to see how far one can push this approach. The results of Kitsunai, Kobayashi, and Tamaki [36] that triples are also directed pathwidth obstacles for h -semicomplete digraphs is a further step in this direction. By Theorem 4 and Equation 1 however, the class of h -semicomplete digraphs is essentially as far as it goes, even if we allow for congestion (cf. Theorem 19). To prove Theorem 4, we take inspiration from Slivkins' [41] proof that k -EDGE DISJOINT PATHS is $W[1]$ -hard parameterized by k on DAGs. In his reduction from the CLIQUE instance (G, ℓ) , a DAG in a matrix-like format is built in a way that each row corresponds to a copy of G and each column to a vertex. Moreover, only one cell may be available in each row, and this must correspond to a vertex of G in the clique. Requests are added so that ℓ of them are used to enforce the uniqueness, while $\binom{\ell}{2}$ are used to encode the edges of the clique (cf. Figure 8). The uniqueness of the available cell is the main obstacle to obtain Theorem 4. In Slivkins' proof, it is obtained by placing two parallel paths, named a and b , with a request from the first vertex of a to the last of b , which may only be fulfilled if we perform a jump at the appropriate point: from where the segment of a corresponding to the desired vertex begins to where the corresponding segment in b ends. We adapt his proof by making each

of the paths a and b a tournament (cf. Figure 6). This, however, opens us to cheating: an edge-encoding path could randomly walk along a suffix of a or a prefix of b and break the proof. To overcome this, we require that a and b are completely occupied, except at the segments corresponding to the desired vertex of G ; this is achieved by introducing several structures and requests in order to enforce a synchronous behavior in each row of the matrix (cf. Figure 9). Using a long path strategy, similarly to the one used in Theorem 11, we are able to force that every vertex participates in the fulfillment of several requests while keeping both the congestion high ($c = k/d$ for some constant $1 < d \leq k$) and the directed pathwidth equal to two, proving that a statement analogous to Theorem 4 (i.e., Theorem 19) also holds for (k, c) -DDP.

3 Definitions and preliminaries

In this section we provide basic preliminaries about digraphs (including the definition of directed pathwidth), parameterized complexity, and tournaments and related classes along with the notion of k -triple. In Subsection 3.3 we present and analyze the counterexample given by Fomin and Pilipczuk [23] that shows that the irrelevant vertex technique is not extensible to k -DDP.

3.1 Digraphs, directed pathwidth, and parametrized complexity

For basic background on graph theory we refer the reader to [6]. Since in this article we mainly work with digraphs, we focus on basic definitions of digraphs, often skipping their undirected counterparts. Given a digraph G we denote by $V(G)$ and $E(G)$ the sets of vertices and arcs of G , respectively. All involved digraphs are simple, i.e., they have neither loops nor parallel arcs. Given $X \subseteq V(G)$, we denote by $G \setminus X$ the digraph resulting from removing every vertex of X from G . We denote by $G[X]$ the subgraph of G induced by X .

If e is an arc of a digraph from a vertex u to a vertex v , we say that e has *endpoints* u and v , that e is *incident* to u and v , and that e is *oriented* from u to v . We may refer to e as the ordered pair (u, v) . In this case, u is the *tail* of e and v is the *head* of e . We also say that e is *leaving* u and *reaching* v , and that u and v are *adjacent*. A *clique* in a (di)graph G is a set of pairwise adjacent vertices of G , and an *independent set* is a set of pairwise non-adjacent vertices of G . A pair of arcs is adjacent if they share an endpoint. A *matching* is a set of pairwise non-adjacent arcs. For $A, B \subseteq V(G)$, we say that M is a *matching from A to B* if every arc of M has tail in A and head in B . The *in-degree* (resp. *out-degree*) of a vertex v in a digraph G is the number of arcs with head (resp. tail) v . We denote the in-degree and out-degree of v by $\deg_G^-(v)$ and $\deg_G^+(v)$, respectively.

The *in-neighborhood* $N_D^-(v)$ of v is the set $\{u \in V(D) \mid (u, v) \in E(G)\}$, and the *out-neighborhood* $N_D^+(v)$ is the set $\{u \in V(D) \mid (v, u) \in E(G)\}$. We say that u is an *in-neighbor* of v if $u \in N_D^-(v)$ and that u is an *out-neighbor* of v if $u \in N_D^+(v)$. We extend these notations to sets of vertices: given $X \subseteq V(D)$, we define $N_D^-(X) = (\bigcup_{v \in X} N_D^-(v)) \setminus X$ and $N_D^+(X) = (\bigcup_{v \in X} N_D^+(v)) \setminus X$.

A *walk* in a digraph G is an alternating sequence W of vertices and arcs that starts and ends with a vertex, and such that for every arc (u, v) in the walk, vertex u (resp. vertex v) is the element right before (resp. right after) arc (u, v) in W . If the first vertex in a walk is u and the last one is v , then we say this is a *walk from u to v* . A *path* on p vertices is a digraph formed by p pairwise distinct vertices; to avoid confusion with sets of vertices, $P = \langle v_1, \dots, v_p \rangle$ always denotes the path P starting at v_1 , ending at v_p , following the arcs in the given order of the vertices. We denote by \prec_P the *comes before than in P* relation, i.e.,

$v_i \prec_P v_j$ if and only if $i < j$. When the ordering itself is unimportant, we use $V(P)$ to refer to its vertex set.

Given $X, Y \subseteq V(D)$, we say that X is *complete to* Y if $(u, v) \in E(D)$ whenever $u \in X$ and $v \in Y$.

For a positive integer k , we denote by $[k]$ the set of integers $\{1, \dots, k\}$.

Directed pathwidth and k -triples. A *path decomposition* of a digraph D is a sequence (X_1, \dots, X_p) of vertex subsets of D , called *bags*, such that the following three conditions are satisfied:

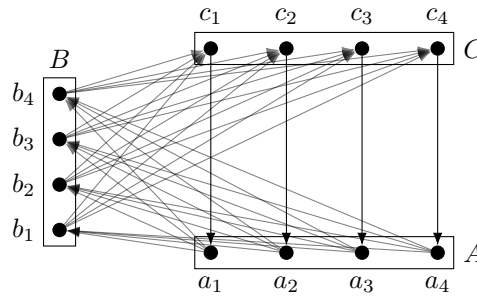
1. $\bigcup_{i \in [p]} X_i = V(D)$,
2. for every arc $(u, v) \in E(D)$, we either have $u, v \in X_i$ for some X_i , or $u \in X_i$ and $v \in X_j$ for some $i \geq j$, and
3. for every vertex $v \in V(D)$, the set $\{i \mid v \in X_i\}$ of indices of the bags containing v forms a single integer interval.

Similarly to undirected pathwidth, the *width* of a path decomposition (X_1, \dots, X_p) of a digraph D is defined as $\max_{i \in [p]} |X_i| - 1$, and the *directed pathwidth* of D , denoted by $\text{dpw}(D)$, is the smallest integer ℓ such that there exists a path decomposition of D of width ℓ .

► **Definition 5 (k -triple).** Let D be a digraph. For an integer $k \geq 1$, a k -triple \mathcal{K} of D is formed by an ordered triple (A, B, C) of disjoint subsets of $V(D)$ and

- $|A| = |B| = |C| = k$; and
- there are orderings (a_1, \dots, a_k) , (b_1, \dots, b_k) , and (c_1, \dots, c_k) of A, B , and C , respectively, such that
 - for all $i, j \in [k]$ we have $(a_i, b_j), (b_i, c_j) \in E(D)$, and
 - for all $i \in [k]$ we have $(c_i, a_i) \in E(D)$.

When working with a k -triple (A, B, C) , it is useful to have an easy way to refer the associated endpoints of the matching from C to A . Thus we sometimes refer to this matching as a bijective mapping M from C to A . That is, for $i \in [k]$ we have $M(c_i) = a_i$ and $M^{-1}(a_i) = c_i$. In addition, we extend graph-theoretical notation to k -triples in the following way. If \mathcal{K} is a k -triple, we denote by $V(\mathcal{K})$ the set $A \cup B \cup C$ and by $E(\mathcal{K})$ the set of all arcs between pairs of vertices of $V(\mathcal{K})$. See Figure 1 for an example of a 4-triple.



■ **Figure 1** A 4-triple (A, B, C) . We remark that nothing is known about the arcs inside A , B , or C nor about arcs between C and A other than the ones in the matching.

Parameterized complexity. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, for some finite alphabet Σ . For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, the value k is called the *parameter*. Such a problem is *fixed-parameter tractable* (FPT for short) if there is an algorithm that decides membership in L of an instance (x, k) in time $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable

	\mathcal{T}	\mathcal{S}	\mathcal{C}_h
k -DDP	NP-c. [Theorem 1]	–	–
(k, c) -DDP	NP-c. for $c = k^\varepsilon$, $\varepsilon \in [0, 1)$ [Theorem 3]	FPT by $k + h$ when $2c > k$ [Theorem 2]	W[1]-h. by $k + h$ with $\text{dpw}(T) \leq 2$ [Theorem 4]

■ **Table 2** Summary of our main results.

function f , and it is in the class XP if there is an algorithm that decides membership in L of an instance (x, k) in time $f(k) \cdot |x|^{g(k)}$ for some computable functions f and g . Consult the monographs [17, 19, 21, 22, 39] for background on parameterized complexity.

3.2 Tournaments and relatives

As already said in the introduction, we denote by \mathcal{T} and \mathcal{S} the classes of tournaments and semicomplete digraphs, respectively. For every non-negative integer h , we consider three classes of digraphs. The class \mathcal{S}_h contains every digraph D such that every $v \in V(D)$ has at most h non-neighbors. The class \mathcal{C}_h contains every digraph whose vertex set can be partitioned into at most h cliques. The class \mathcal{A}_h contains every digraph D with $\alpha(D) \leq h$.

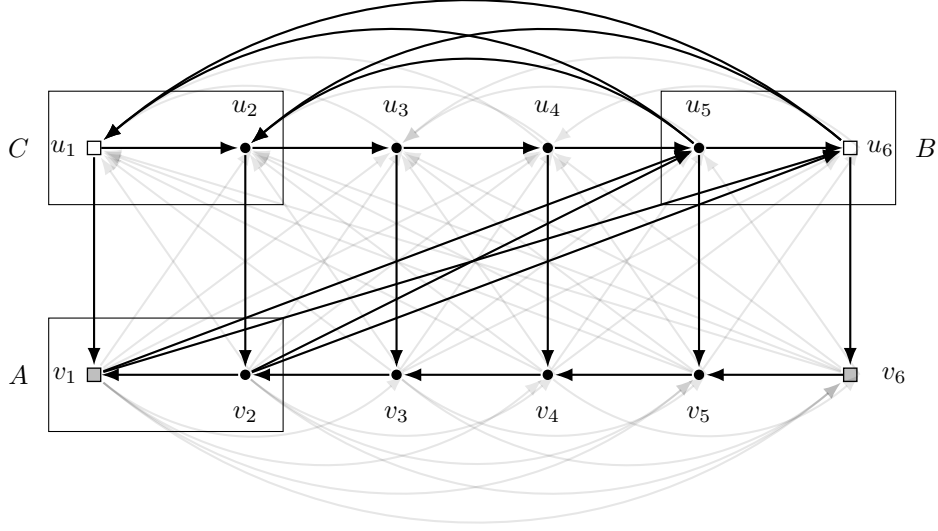
Brook's Theorem [8] states that the vertex set of every graph G with maximum degree $\Delta(G) \leq k$ can be partitioned into $k + 1$ color classes X_1, \dots, X_{k+1} such that each X_i induces an independent set in G . Let $D \in \mathcal{S}_h$. Applying Brook's Theorem to the complement \overline{D} of D , we conclude that $V(D)$ can be partitioned into $h + 1$ cliques and thus $\mathcal{S}_h \subseteq \mathcal{C}_{h+1}$. Since every independent set of a digraph intersects any clique in at most one vertex, it also holds that $\mathcal{C}_h \subseteq \mathcal{A}_h$. In Table 2 we give a summary of our main results.

3.3 Analyzing the Fomin-Pilipczuk counterexamples

Let us generalize the counterexamples given in [23] to other values of the congestion; this will show that the irrelevant vertex technique is not extensible to (k, c) -DDP for appropriate values of c . For each $n \geq 1$ and $c \geq 1$, we build an instance (T_n, K, c) of (k, c) -DDP in tournaments such that:

1. The digraph T_n has $4(n + 1)$ vertices.
2. The vertex set of T_n is partitioned into two sets U, V of the same size, with $U = \{u_i \mid i \in [2n + 2]\}$ and $V = \{v_i \mid i \in [2n + 2]\}$.
3. The tournament $T_n[U]$ has a path $P_u = \langle u_1, \dots, u_{2n+2} \rangle$ and all other arcs going in the opposite direction. Similarly, $T_n[V]$ has a path $P_v = \langle v_{2n+2}, \dots, v_1 \rangle$ and all other arcs going in the opposite direction.
4. For every pair $(u_i, v_j) \in U \times V$, arc (u_i, v_j) exists if $i = j$, otherwise we have the arc (v_j, u_i) .
5. Digraph T_n has an n -triple where $A = \{v_1, v_n\}$, $B = \{u_{n+3}, \dots, u_{2n+2}\}$, and $C = \{u_1, \dots, u_n\}$.
6. There are c requests (u_1, u_{2n+2}) and c requests (v_{2n+2}, v_1) .

We refer to Figure 2 for the case $n = 2$. Observe that the requests can only be satisfied by using P_u and P_v , c times each. This statement follows by induction on n : it suffices to observe that u_2 is the unique vertex that may be the second vertex in the (u_1, u_{2n+2}) -satisfying paths, while v_2 is the unique vertex that may be the penultimate vertex in the paths that



■ **Figure 2** Counterexample for $n = 2$, where non-black vertices of the same color correspond to endpoints of a same request. The rectangles indicate the three sets that make up the n -triple.

satisfy (v_{2n+2}, v_1) ; at this point we can discard u_1, v_1 and repeat the analysis. We can further restrict ourselves to congestion values smaller than $k/2$. For example, if we add τ copies of T_n and add all arcs from the i -th copy to the j -th whenever $i < j$, then we can get $c = k/(2\tau)$; it is not hard to come up with other strategies that would yield $c = k/d$ for every $d \in [2, k]$. At this point, it is natural to think about the interval $(1, 2)$? Interestingly, things break down in this case. Suppose that, instead of having the same number of requests between the endpoints of P_u and P_v , we had two requests (u_1, u_{2n+2}) , only *one* request for (v_{2n+2}, v_1) , and congestion $c = 2$. We could use P_u twice and P_v once, but we could also do the following: begin by satisfying (v_{2n+2}, v_1) using P_v ; now, add the paths $\langle u_1, v_1, u_6 \rangle$ and $\langle u_1, u_2, v_2, u_{2n+2} \rangle$ to satisfy the (u_1, u_{2n+2}) requests. With this, we have avoided the usage of every vertex in $\{u_3, \dots, u_{2n+1}\}$, making them *irrelevant* to the instance; we could also have avoided using vertices in V using extra steps.

4 Hardness results

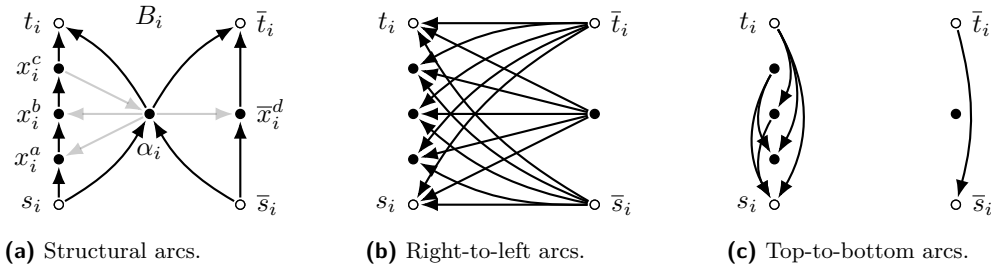
While it had been assumed for a long time that DIRECTED DISJOINT PATHS was an NP-hard problem on tournaments [1, 14, 33, 41], we verified that there is a mistake in the proof given in [2]. We have reached out to its authors, which have confirmed the flaw in their proof in a personal communication [3]. In this section, we provide a direct NP-hardness proof for tournaments as well as other lower bounds the C_h superclasses.

4.1 Tournaments

Fortunately, as we show in the following theorem, k -DDP is indeed NP-hard on tournaments. We highlight that our reduction is completely different from the approach in [2]. In particular, we reduce from (3,1)-3-SAT, a variant of 3-SAT where each variable appears exactly four times – once negated and three times unnegated – and that was shown to be NP-complete

in [18].

Construction. Let (X, \mathcal{C}) be the input instance to (3,1)-3-SAT, where X is the set of variables and \mathcal{C} is the set of clauses, and (T, K) be the k -DDP instance we are going to build, with T denoting the digraph, and K denoting the set of requests. We begin by picking an arbitrary but fixed order $\langle x_1, \dots, x_n \rangle$ of X . Now, for each variable $x_i \in X$, add a copy of the *directed butterfly gadget* shown in Figure 3 to T , while (s_i, t_i) and (\bar{s}_i, \bar{t}_i) are added to K ; we denote this gadget by B_i . Intuitively, the paths between the white vertices of B_i must be contained in it and, since the gadget has only once center (namely, α_i in Figure 3), at most one of them may avoid the “wings”, as shown in Observation 7. This allows us to encode the assignment of a variable and only satisfy clauses for which the appropriate literal is true, i.e., adding $\langle s_i, \alpha_i, t_i \rangle$ to the solution is equivalent to setting $x_i = \text{true}$.



■ **Figure 3** Directed butterfly gadget for variable x_i , occurring negated in clause C_d and unnegated in clauses C_a, C_b , and C_c . White vertices represent terminals. The given orientations of the gray arcs will be important when talking about congested versions of the problem. Vertex α_i is the *center* of B_i , while the two disjoint paths from the s vertices to the t vertices that do not use α_i are its *wings*.

After building all n butterflies, add an arc from every $u \in B_j$ to every $v \in B_i$ whenever $j > i$. To encode our clauses, take each $C_a \in \mathcal{C}$, add new vertices p_a, q_a to T , the pair (p_a, q_a) to K , and the arc (q_a, p_a) ; arcs between each p_a and q_b , for $a \neq b$, can be added arbitrarily. Now, take $a \in \{1, \dots, m\}$ as an example, and suppose that the a -th clause of \mathcal{C} is $C_a = (x_1 \vee x_2 \vee \bar{x}_3)$; in this case, we add arcs from p_a to the vertices x_1^a, x_2^a , and \bar{x}_3^a , and from these to q_a . For simplicity, the superscript j of each x_i^j and \bar{x}_i^j is the same as the subscript of the corresponding clause C_j . At this point, the only missing arcs to ensure we have a tournament are between vertices of butterfly gadgets and clause gadgets; we add them such that p_a has no other outgoing arc and q_a has no other incoming arc. This concludes the construction of (T, K) , with $K = \{(s_i, t_i), (\bar{s}_i, \bar{t}_i) \mid x_i \in X\} \cup \{(p_a, q_a) \mid C_a \in \mathcal{C}\}$. For the above example, our goal is to witness the satisfiability of C_a with a path $\langle p_a, y, q_a \rangle$, where $y \in \{x_1^a, x_2^a, \bar{x}_3^a\}$, which will have to be the case as $N^+(p_a) = N^-(q_a) = \{x_1^a, x_2^a, \bar{x}_3^a\}$.

► **Lemma 6.** *If there is a satisfying assignment π for (X, \mathcal{C}) , there is a family of disjoint paths \mathcal{P} satisfying (T, K) .*

Proof. We break our analysis in two. First, for each variable $x_i \in X$, if $\pi(x_i) = 0$, then add the paths $\langle \bar{s}_i, \alpha_i, \bar{t}_i \rangle$ and $\langle s_i, x_1^a, x_1^b, x_1^c, t_i \rangle$ to \mathcal{P} ; otherwise add $\langle s_i, \alpha_i, t_i \rangle$ and $\langle \bar{s}_i, \bar{x}_i^d, \bar{t}_i \rangle$ to \mathcal{P} . Now, let C_a be a clause and ℓ_i one of its satisfying literals. If $\ell_i = \bar{x}_i$, then $\pi(x_i) = 0$ and, by our choices in B_i , vertex \bar{x}_i^a is not used by any path in \mathcal{P} , so we may now add $\langle p_a, \bar{x}_i^a, q_a \rangle$ to \mathcal{P} . A similar choice can be performed if $\ell_i = x_i$; note that, for each vertex in the left wing of B_i , a is the unique index such that $(p_a, x_i^a), (x_i^a, q_a) \in E(T)$, so no other clause gadget could have used x_i^a in its satisfying path. ◀

► **Observation 7.** Let B_i be one of the butterfly gadgets of (T, K) . If (T, K) admits a solution \mathcal{P} , then (i) the paths that satisfy (\bar{s}_i, \bar{t}_i) and (s_i, t_i) are internal to B_i , and (ii) at least one of the wings of B_i is entirely occupied by such a path.

Proof. Let $\bar{P}_i, P_i \in \mathcal{P}$ be the paths connecting (\bar{s}_i, \bar{t}_i) and (s_i, t_i) , respectively. Suppose that \mathcal{P} is minimal, i.e., none of its paths has an internal shortcut; for example, $\langle s_i, \alpha_i, \bar{t}_i, t_i \rangle \notin \mathcal{P}$ because arc (α_i, t_i) has both endpoints contained in the path and could be used to shorten it. Let us first show that $\bar{P}_i \subseteq B_i$. Recall that $N^-(\bar{t}_i) \subseteq \{\bar{x}_i^d, \alpha_i\} \cup \{v \mid v \in B_j, j > i\}$ and that no terminal of $V(K)$ can be used as an intermediate vertex of a path, as they must be saturated by their own paths, so the only arcs incident to \bar{t}_i that may be used to reach it are the ones we have just listed. As such, the penultimate vertex of \bar{P}_i is either one of $\{\bar{x}_i^d, \alpha_i\}$, in which case we are done as $\{\bar{x}_i^d, \alpha_i\} \subseteq N^+(\bar{s}_i)$ and \mathcal{P} is minimal, or it is in some B_j with $j > i$, but this is impossible as there is no path from \bar{s}_i to such a vertex in $G \setminus V(K)$ by design. A symmetric analysis can be performed to show that $P_i \subseteq B_i$: if $\alpha_i \in P_i$, we are done as $\alpha_i \in N^+(s_i) \cap N^-(t_i)$, so suppose $\alpha_i \notin P_i$. Note that B_j , with $j > i$ is unreachable from s_i in $G \setminus V(K)$, and so we have that $P_i \cap B_j = \emptyset$. Moreover, for $\ell < i$, t_i is unreachable from any vertex of B_ℓ in $G \setminus V(K)$. Finally, every path between s_i and the right wing of B_i in $G \setminus V(K)$ passes through α_i , which is not in P_i , so the right wing of B_i does not intersect P_i . Consequently, if $\alpha_i \notin P_i$, then $P_i \subseteq B_i \setminus \{\alpha_i, \bar{s}_i, \bar{x}_i^d, \bar{t}_i\}$, which is precisely the left wing of B_i , completing the proof of (i). Towards proving (ii), if $P_i = \langle s_i, x_i^a, x_i^b, x_i^c, t_i \rangle$ we are done, so suppose that $P_i = \langle s_i, \alpha_i, t_i \rangle$; by (i), this implies that $\bar{P}_i = \langle \bar{s}_i, \bar{x}_i^d, \bar{t}_i \rangle$. A completely symmetric analysis can be performed if $\bar{P}_i = \langle \bar{s}_i, \alpha_i, \bar{t}_i \rangle$ to conclude that $P_i = \langle s_i, x_i^a, x_i^b, x_i^c, t_i \rangle$. ◀

► **Lemma 8.** If there is a family of disjoint paths \mathcal{P} satisfying (T, K) , there is a satisfying assignment π for (X, \mathcal{C}) .

Proof. Let $\bar{P}_i, P_i \in \mathcal{P}$ be the paths satisfying (\bar{s}_i, \bar{t}_i) and (s_i, t_i) , respectively. By [Observation 7](#), at least one of these two paths occupies a wing. If $\bar{P}_i = \langle \bar{s}_i, \bar{x}_i^d, \bar{t}_i \rangle$, we set $\pi(x_i) = 1$, otherwise set $\pi(x_i) = 0$. To see that this is a satisfying assignment, take $C_a \in \mathcal{C}$ and the corresponding path $P_a = \langle p_a, v_i, q_a \rangle$, such that $v_i \in B_i$. If $v_i = x_i^a$, then $x_i \in C_a$ and $x_i^a \notin P_i$ as $P_i \cap P_a = \emptyset$, but this means that $P_i = \langle s_i, \alpha_i, t_i \rangle$ as this is the unique path that can satisfy the request (s_i, t_i) and not use x_i^a by [Observation 7](#). The latter in turn implies that $\bar{P}_i = \langle \bar{s}_i, \bar{x}_i^d, \bar{t}_i \rangle$, $\pi(x_i) = 1$, and so C_a is satisfied. If, however, $v_i = \bar{x}_i^a$, then $\bar{x}_i \in C_a$, and $\bar{P}_i = \langle \bar{s}_i, \alpha_i, \bar{t}_i \rangle$, in which case we have set $\pi(x_i) = 0$, satisfying C_a . ◀

► **Theorem 1.** DIRECTED DISJOINT PATHS on tournaments is NP-complete.

Proof. This follows immediately from the above construction and lemmas, as well as the straightforward algorithm that, given a solution \mathcal{P} to (T, K) , checks whether \mathcal{P} satisfies the requests K and its paths are pairwise disjoint. ◀

4.2 Tournaments with congestion

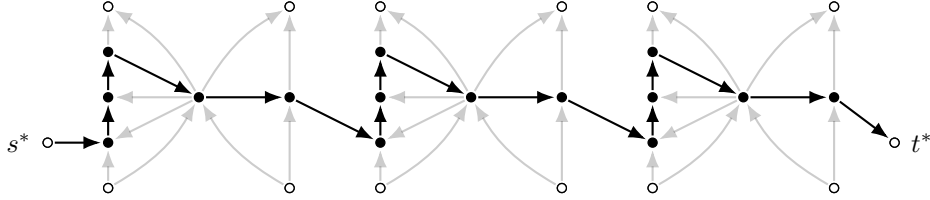
As our main focus for this work is on congested variants of DIRECTED DISJOINT PATHS, it is natural to extend the proof of [Theorem 1](#) to these variants as well. In particular, we are interested in the case where the congestion $c = |K|/d$.

4.2.1 Restricted K

We first consider a variant that we call RESTRICTED (k, c) -DDP, where we are additionally forbidden from using vertices of K as inner vertices of other paths; that is, if $(u, v) \in K$,

then a path P has $u, v \in P$ if and only if u is the first vertex of P and v is the last. Note that if $c = 1$, this is precisely what we have in the standard k -DDP problem.

Meta-construction. We reduce from the instance (T, K) of k -DDP built in the proof of Theorem 1 to construct the instance (T', K', c) of RESTRICTED (k', c) -DDP. Let $\langle B_1, \dots, B_n \rangle$ be the ordering of the butterfly gadgets, that is, B_j is complete to B_i for all $j > i$. We perform the following modification to T : for each $i \in [n-1]$, reverse the arc (x_{i+1}^a, \bar{x}_i^d) , where x_{i+1}^a is the unique out-neighbor of s_{i+1} in the left wing of B_{i+1} . To conclude the construction of T' , we add two vertices s^*, t^* and arcs such that x_1^a is the unique out-neighbor of s^* and \bar{x}_n^d is the unique in-neighbor of t^* . Finally, we add $c-1$ requests (s^*, t^*) and one request (t^*, s^*) to K , thus obtaining K' . Note that c is *almost* arbitrary, as it can be set to a constant or whichever *fraction* of $|K'|$ that we wish. Not every element of $o(|K'|)$, however, is assignable to c ; for example, $c = |K'| - \log |K'|$ is not a viable candidate, as this would imply that $|K'|$ is exponential in $|K|$ since $|K'| = |K| + c$. We illustrate the resulting tournament T' in Figure 4.



■ **Figure 4** Queuing of the butterfly gadgets performed for the construction of T' . Gray arcs do not participate in the critical path of T' , while black arcs do. Again, within the butterfly gadgets, we add the missing arcs from top-to-bottom, right-to-left. All other arcs are also right-to-left.

Observe that the interior vertices of the butterflies' wings are the only vertices of T' that do not participate in any element of K' . Moreover, by construction, they all participate in the unique s^* - t^* path of $T' \setminus V(K' \setminus \{(s^*, t^*)\})$ in the same order specified by $\langle B_1, \dots, B_n \rangle$. We call this path the *critical path* of T' , and in any solution to (T', K', c) , all of its inner vertices participate in at least $c-1$ paths; cf. the black arcs in Figure 4.

► **Theorem 9.** RESTRICTED (k', c) -DDP *remains NP-complete even when restricted to tournaments and when $c = \varepsilon k$ for every $\varepsilon \in (0, 1)$.*

Proof. Given the desired fraction $c = |K'|/d$, we begin by restricting ourselves to instances of k -DDP such that $|K| - 1$ is divisible by $d - 1$, and then constructing $(T', K', c = |K'|/d)$ as above. Now, note that, since the critical path of T' must be occupied by $c-1$ paths from s^* to t^* , the remaining requests of K' in T' are exactly the same requests of K in T with the exact same availability for each vertex in $V(T') \cap V(T)$: terminals of K can only be used by themselves in both problems, while each non-terminal can either be occupied by one request between butterfly terminals or by one request from outside a butterfly gadget. As such, it follows from Theorem 1 that (T, K) admits a solution if and only if (T', K', c) admits a solution. ◀

4.2.2 Unrestricted K

While we are unable to prove a statement as strong as Theorem 9 for (k, c) -DDP, we are able to (conditionally) rule out polynomial-time algorithms for some choices of c ; in particular, if $c \leq |K|^\varepsilon$ for $\varepsilon \in [0, 1)$ we can show such a lower bound.

► **Theorem 3.** (k, c) -DDP remains NP-complete on tournaments even restricted to instances satisfying $c = k^\varepsilon$, for every $\varepsilon \in [0, 1)$.

Proof. The construction is simple: let (T', K', c') be an instance of RESTRICTED (k', c') -DDP obtained in the proof of Theorem 9 and (T, K, c) be the instance of (k, c) -DDP we are building. We initially set $T \leftarrow T'$, $K \leftarrow K'$, and $c \leftarrow c'$. Recall that there are exactly c requests with s^*, t^* as both of its endpoints and all other terminals are endpoints of exactly one request. Now, let X be the set of all requests of the latter type, and, for each $(u, v) \in X$, add $c - 1$ requests of the form (v, u) to K . As such, we have that $|K| = |K'| + |X|(c - 1) = (c + |X|) + |X|(c - 1) = (|X| + 1)c$ and, consequently, $|K|/c = |K|^{1-\varepsilon} = |X| + 1$ is polynomial in $|X|$.

The correctness of the reduction follows immediately from the fact that each terminal in (T, K, c) must be completely occupied by paths that have it as an endpoint, and so all that remains to be solved is the original RESTRICTED (k', c') -DDP instance. ◀

We would like to point out that overcoming the $|K|^\varepsilon$ barrier has proved extremely challenging for us, and seems to require a very different approach to the one we employed so far. In particular, we can rearrange the butterfly gadgets to be *stacked* instead of *queued*, i.e., we glue the terminals of two butterflies, avoiding the need of too many paths to saturate them. This, however, is not enough, as we are unable to saturate the clause gadgets with few paths, instead always requiring $\mathcal{O}(|\mathcal{C}|)$ requests, as done in Theorem 3. Interestingly, we can prove a weak NP-hardness result for larger values of the congestion, namely if $c = |K| \log^{-\varepsilon} |K|$.

► **Theorem 10.** (k, c) -DDP is weakly NP-hard on tournaments if the congestion is of the form $|K| \log^{-\varepsilon} |K|$, for every $\varepsilon > 0$.

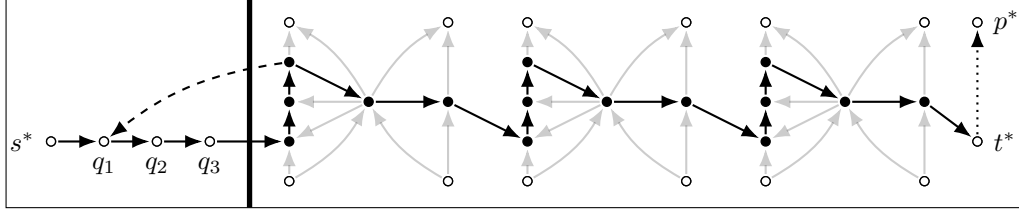
Proof. The construction and correctness arguments are precisely the same as the ones in the proof of Theorem 3. Instead of opting for an explicit representation of K , we instead encode it as map $K : V(T) \times V(T) \mapsto \mathbb{N}$, where $K(u, v)$ represents the number of requests from vertex u to vertex v . Since $c = |K| \log^{-\varepsilon} |K|$, it follows that $\log^\varepsilon |K| = |X| + 1$ and, consequently, each $K(u, v)$ can be encoded in $(|X| + 1)^{1/\varepsilon}$ bits. ◀

4.2.3 NP-hardness for graphs in \mathcal{C}_2

Before proceeding, recall that $G \in \mathcal{C}_h$ if and only if the vertices of its underlying graph can be partitioned by at most h cliques, i.e., its complement has chromatic number at most h . Our construction is similar to the one employed on Theorem 9 and yields an equivalent statement, however there are some key differences, so we do not use it as directly as in Theorem 3. As such, we describe our reduction directly from (3,1)-3-SAT. Intuitively, we will use the flexibility of some non-arcs to extend the critical path with some sinks of the clause requests; at the same time, we can replace the $\mathcal{O}(|T|)$ sources with a single vertex.

Construction Let once again (X, \mathcal{C}) be the given (3,1)-3-SAT instance and take an arbitrary ordering $\langle C_1, \dots, C_m \rangle$ of \mathcal{C} . We first show how to construct an instance (G, K, c) of DIRECTED CONGESTED DISJOINT PATHS with $c = |\mathcal{C}|$, and subsequently show how to extend it to so c is an arbitrary ratio of K . Initially, add the queued structure of butterflies shown in Figure 4 to G and the corresponding requests to K , but now we perform some modifications to it: (i) take the unique out-neighbor v of s^* and subdivide the arc (s^*, v) exactly $|\mathcal{C}|$ times, assigning to each generated q_a a unique clause C_a ; (ii) for each C_a , add an arc from x_i^a to q_a if $x_i \in C_a$ and from \bar{x}_i^a to q_a if $\bar{x}_i \in C_a$; (iii) for each pair $a, b \in [m]$, add the arc (q_b, q_a) if and only if $b > a$; (iv) now, add a unique vertex p^* to G , which has all vertices in the butterfly gadgets as out-neighbors and t^* as unique in-neighbor. Finally, add the arc and the request (t^*, s^*) ,

$c - 1$ requests of the form (s^*, t^*) and, for each $C_a \in \mathcal{C}$, add the request (p^*, q_a) to K . We exemplify the above changes in Figure 5, where the rectangles with the thick vertical line as one of the sides correspond to the cliques partitioning G .



■ **Figure 5** Queuing of the butterfly gadgets performed for the construction of G with the extended critical path on the left of the figure. Each of the two smaller rectangles are cliques in the host graph. Within them, missing arcs are from top-to-bottom, right-to-left. Across the thick vertical line, the only arcs are from vertices of the form $\ell_i^a \in B_i$ to q_a , which exists if and only if $\ell_i \in C_a$; these are shown by the dashed arc from the first butterfly to q_1 . The dotted arc is not considered part of the critical path. Arc $(t^*, s^*) \in E(G)$ is omitted for simplicity.

► **Theorem 11.** (k, c) -DDP is NP-complete when restricted to graphs of \mathcal{C}_2 .

Proof. The forward direction should follow immediately from the construction and the arguments used in the proof of Lemma 6, with the additional detail that the critical path is used to satisfy all $c - 1$ requests from s^* to t^* . For the converse, note that the critical path remains the unique path from s^* to t^* , and so every vertex of Figure 5 incident to a continuous black edge must always be occupied by at least $c - 1$ paths. Moreover, no other path may use q_a as an intermediate vertex since q_a is also an endpoint of the request $(p^*, q_a) \in K$. As such, we may assume that every path satisfying a request originating from p^* is of the form $\langle p^*, \ell_i^a, q_a \rangle$, as the only way to reach a q_a is through some ℓ_i^a and ℓ_i^a is an out-neighbor of p^* . Consequently, we have our assignment π for (X, \mathcal{C}) : set $\pi(x_i) = 1$ if and only if $\{s_i, \bar{x}_i^d, t_i\}$ is the path satisfying $(s_i, t_i) \in K$. ◀

It is also possible to obtain similar proofs when setting $c < |\mathcal{C}|$. The proof follows the same steps as the one of Theorem 11, but requires splitting p^* into sufficiently many vertices to accommodate the fewer requests that can touch p^* . For example, if we want $c = |\mathcal{C}|/2$ and this is an integer value, then we replace p^* with p_1^*, p_2^* and evenly divide the requests of the form (p^*, q_a) between them. We omit the more technical details for brevity.

► **Theorem 12.** (k, c) -DDP is NP-complete when restricted to graphs of \mathcal{C}_2 for every constant congestion value $c \geq 1$.

We conclude our polynomial-time lower bounds with the following theorem.

► **Theorem 13.** (k, c) -DDP is NP-complete when restricted to graphs of \mathcal{C}_2 even if the congestion satisfies $c = \lfloor K \rfloor / d$ for $d > 1$.

Proof. Observe that the construction of Theorem 11 gives us $|K| = 2n + m + c = 2n + 2m$, where $n = |X|$ and $m = |\mathcal{C}|$. To obtain c as a fraction of $|K|$, it suffices to add, for each unit of congestion that exceeds m , one request from s^* to p^* . This request must be satisfied by picking the entire critical path since t^* is the unique in-neighbor of p^* . As such, $|K| = 2n + 2m + z$, and we may pick $z = c - m$ so that $|K|/c$ is the desired ratio. For example, if we wish for $|K|/c = 2$, then we must pick $z = 2n$, since $|K| = 2n + m + c$ this means that $c = 2n + m = \lfloor K \rfloor / 2$. ◀

4.3 $W[1]$ -hardness on $k + h$ on graphs of bounded directed pathwidth

While our previous lower bounds were all in the polynomial-time world, a natural question is what happens if the numbers $k = |K|$ of requests and h of partitioning cliques are now taken as parameters. In [14], Chudnosvsky, Scott, and Seymour showed a $|V(G)|^{\mathcal{O}((hk)^5)}$ -time algorithm for DIRECTED DISJOINT PATH on \mathcal{C}_h . We show that their XP algorithm cannot be significantly improved by proving that DIRECTED CONGESTED DISJOINT PATHS remains $W[1]$ -hard when parameterized by $k + h$ on graphs of bounded directed pathwidth. Our reduction is heavily inspired by, but requires significant increments to, the proof of Slivkins [41] of the $W[1]$ -hardness of EDGE DIRECTED DISJOINT PATHS on DAGs, where the source problem was CLIQUE parameterized by the solution size; to simplify a bit our arguments, we opt to reduce from MULTICOLORED CLIQUE parameterized by the solution size. As in Section 4.1, we first deal with the completely disjoint version of the problem and then show how to extend to the congested case.

Construction. Let (G, \mathcal{V}) be the input instance to MULTICOLORED CLIQUE where $\mathcal{V} = \{V_1, \dots, V_q\}$ are the color classes and (D, K) denotes the DIRECTED DISJOINT PATHS instance we are going to build. For simplicity, assume that every $V_i \in \mathcal{V}$ has the same size n . The vertices of D are partitioned in a $q \times (2 + n)$ matrix-like fashion, with $D_{i,j} \subseteq V(D)$ corresponding to the j -th vertex of $V_i \in \mathcal{V}$ and $D_{i,0}, D_{i,n+1}$ with no correspondence. For every $i \in [q]$, let us define and add the following sets of vertices to G , where each $D_{i,j}$ is a set of $2q + 3$ vertices that will be specified later:

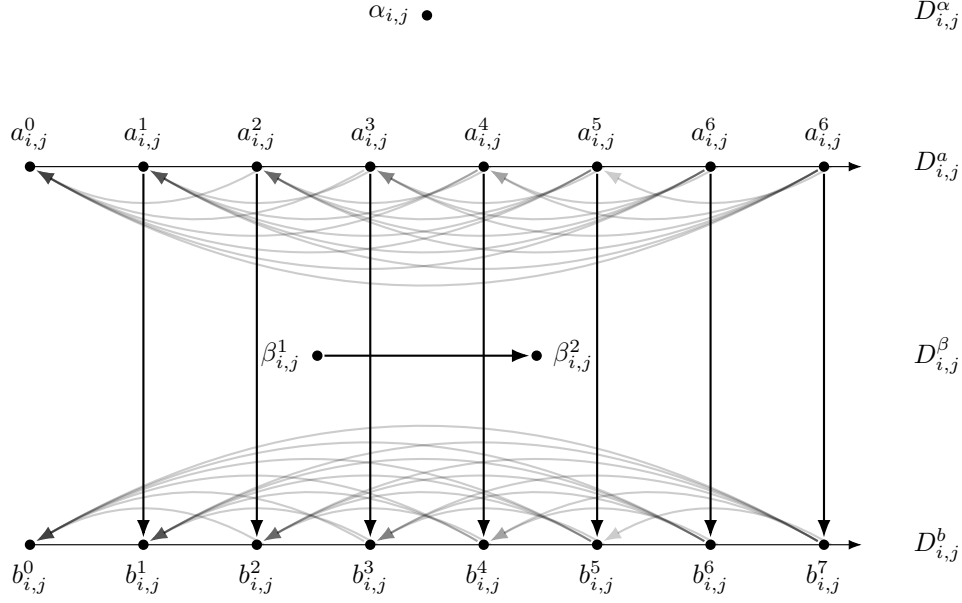
$$\begin{aligned} D_i = & \{\alpha_{i,s}, \alpha_{i,t}, a_{i,s}, a_{i,t}, b_{i,s}, b_{i,t}\} \\ & \cup \{g_{i,s}^1, g_{i,t}^1, g_{i,s}^2, g_{i,t}^2\} \\ & \cup \bigcup_{j \in \{0\} \cup [n+1]} D_{i,j}. \end{aligned}$$

For this proof, we will stick to the convention that vertices with an s subscript or t subscript are sources or targets of a request of K , respectively. In particular, K now contains the pairs $(\alpha_{i,s}, \alpha_{i,t})$, $(a_{i,s}, a_{i,t})$, $(b_{i,s}, b_{i,t})$, $(g_{i,s}^1, g_{i,t}^1)$, and $(g_{i,s}^2, g_{i,t}^2)$ for every $i \in [q]$. Each $D_{i,j}$ is further broken down into four tournaments: $D_{i,j}^\alpha, D_{i,j}^a, D_{i,j}^\beta, D_{i,j}^b$. We set $D_{i,j}^\alpha = \{\alpha_{i,j}\}$, $D_{i,j}^\beta = \{\beta_{i,j}^1, \beta_{i,j}^2\}$, and, for $x \in \{a, b\}$, we set $D_{i,j}^x = \{x_{i,j}^0, x_{i,j}^1, \dots, x_{i,j}^q\}$; intuitively, $x_{i,j}^\ell$ is the only vertex that will connect $D_{i,j}$ to vertices of D_ℓ , while $x_{i,j}^0$ is used to synchronize the internal behavior of D_i . Let us divide our construction in steps:

- (i) The arcs of $D_{i,j}$ including $D_{i,0}$ and $D_{i,n}$ are built as follows: each $D_{i,j}^x$, $x \in \{\alpha, a, \beta, b\}$ has exactly one Hamiltonian path – known as the x -path of $D_{i,j}$ – and all other arcs are in the opposite direction; i.e., if $\langle x_{i,j}^0, \dots, x_{i,j}^\ell \rangle$ is the Hamiltonian path of $D_{i,j}^x$, then $(x_{i,j}^z, x_{i,j}^y) \in E(D)$ for every non-consecutive pair z, y with $z > y$; w.l.o.g. we assume the visitation order of the Hamiltonian path is the same as the one we specified in the definition of $D_{i,j}^x$. The only other arcs incident to two vertices of $D_{i,j}$ are a perfect matching from $D_{i,j}^a$ to $D_{i,j}^b$, such that $a_{i,j}^\ell$ is matched to $b_{i,j}^\ell$ for $\ell \in [n]$. We present an example in Figure 6.

Intuitively, the a - and b -paths will be used in the encoding of the MULTICOLORED CLIQUE instance, while the α - and β -paths are used to synchronize the decisions performed in the a - and b -paths.

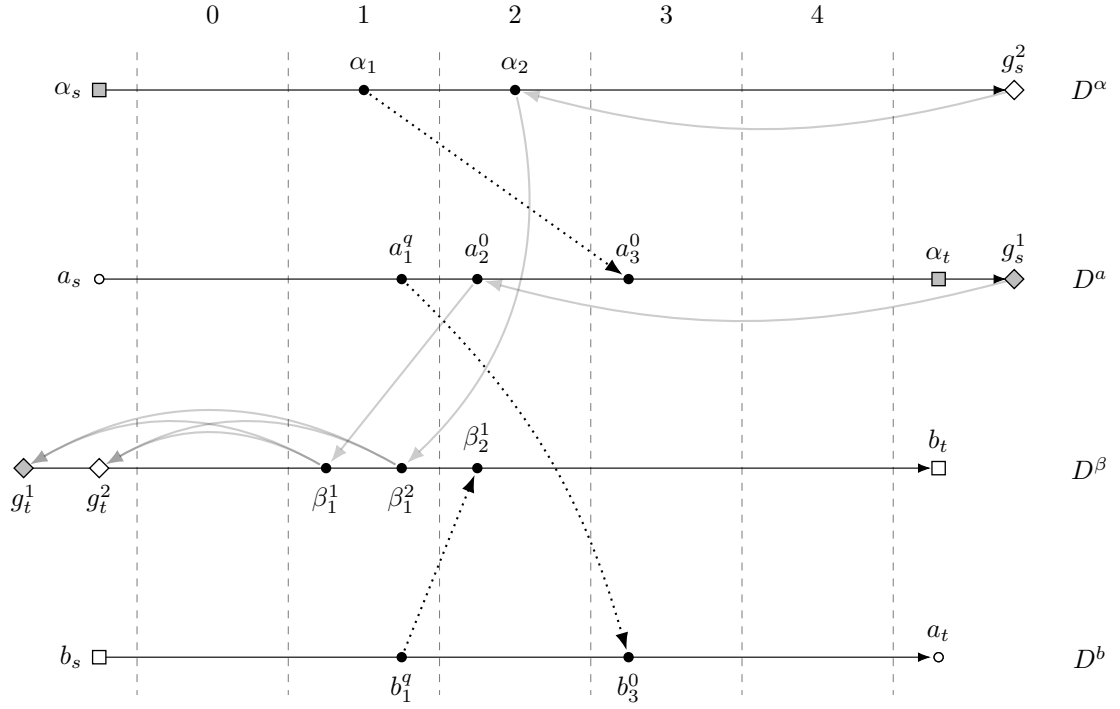
- (ii) The next step in the construction is the connection of the different elements of D_i . Our goal is to add arcs so that exactly one index $j \in [n]$ has both $D_{i,j}^a$ and $D_{i,j}^b$ unoccupied by paths satisfying requests internal to gadget D_i , while all other vertices of D_i^a and D_i^b



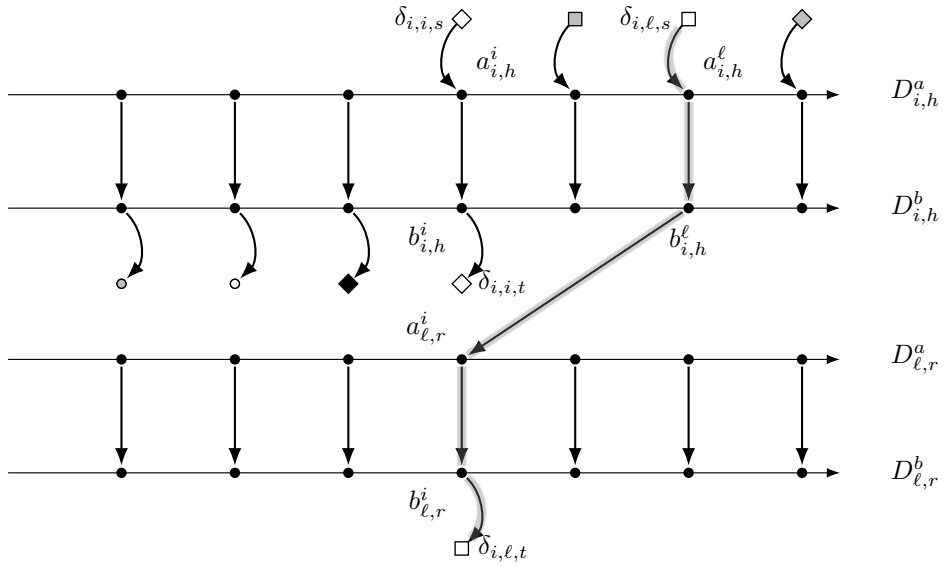
■ **Figure 6** Gadget $D_{i,j}$ and the arcs built in Step (i) of the reduction. Grey arcs are only used to force that their incident vertices form a tournament. Horizontal arrows denote the three x -paths of $D_{i,j}$ with at least one edge, i.e., $x \in \{a, b, \beta\}$.

are occupied; that is, exactly one vertex of each color class, corresponding to that index j , may be picked in the MULTICOLORED CLIQUE instance. This can be accomplished as follows:

- a. For each $j \in [n+1]$ and $x \in \{\alpha, a, \beta, b\}$, take the last vertex of the x -path of $D_{i,j-1}^x$ and add an arc to the first vertex of the x -path of $D_{i,j}^x$, with all other arcs going from $D_{i,j}^x$ to $D_{i,j-1}^x$. Doing this, $D_i^x = D_{i,0} \cup \bigcup_{j \in [n]} D_{i,j}^x$ is a tournament with a unique Hamiltonian path obtained by stitching the x -paths of $D_{i,j}^x$'s to each other.
 - b. Add the arcs $(\alpha_{i,s}, \alpha_{i,0})$, $(a_{i,s}, a_{i,0}^0)$, $(b_{i,s}, b_{i,0}^0)$, the arcs $(a_{i,n+1}^q, \alpha_{i,t})$, $(\beta_{i,n+1}^2, b_{i,t})$, $(b_{i,n+1}^q, a_{i,t})$, and all remaining right-to-left arcs so that the vertices in each long horizontal arrow in Figure 7 form a tournament; e.g. $a_{i,s}$ has $D_i^a \setminus \{a_{i,0}^0\}$ as its in-neighbors while $\alpha_{i,t}$ has $D_i^a \setminus \{a_{i,n+1}^q\}$ as its out-neighbors.
 - c. For each $j \in \{0\} \cup [n-1]$, we add the *forward jumping arcs* $(\alpha_{i,j}, a_{i,j+2}^0)$, $(a_{i,j}^q, b_{i,j+2}^0)$, and $(b_{i,j}^q, \beta_{i,j}^1)$.
 - d. For each $j \in [n]$, we add the *backward jumping arcs* $(\alpha_{i,j}, \beta_{i,j-1}^2)$ and $(a_{i,j}^0, \beta_{i,j-1}^1)$.
 - e. Finally, for each $j \in \{0\} \cup [n]$, we add every arc from $g_{i,s}^1$ to D_i^a , from $g_{i,s}^2$ to D_i^a , from D_i^b to $g_{i,t}^1$ and $g_{i,t}^2$, and between the latter two arbitrarily. Consequently, $\{g_{i,s}^1\} \cup D_i^a$, $\{g_{i,s}^2\} \cup D_i^a$, and $\{g_{i,t}^1, g_{i,t}^2\} \cup D_i^b$ are tournaments.
- (iii) While Step (ii) essentially encodes that only one vertex of each V_i may be picked, we now must make sure that they collectively indeed form a tournament of G . Our final set of vertices are obtained by adding the following requests and their corresponding vertices $(\delta_{i,\ell,s}, \delta_{i,\ell,t})$, where $i \in [q]$ and $\ell \in [q] \setminus [i-1]$; this implies that $|K| = 5q + \binom{q}{2} + q$. We proceed as follows, which is illustrated in Figure 8.
- a. For each triple $i, \ell \in [q], j \in [n]$ with $i \leq \ell$: add the arcs $(\delta_{i,\ell,s}, a_{i,j}^\ell)$, $(b_{i,j}^\ell, \delta_{i,\ell,t})$.
 - b. Finally, let $u \in V_i$ and $v \in V_\ell$ be adjacent vertices of G with $i < \ell$: add the arc $(b_{i,u}^\ell, a_{i,v}^\ell)$.



■ **Figure 7** Gadget D_i and the arcs built in Step (ii) of the reduction. Dotted arcs correspond to the forward jumping arcs of Step (ii.c), while the gray arcs represent the arcs of Steps (ii.d) and (ii.e). We do not illustrate the arcs added in Steps (ii.a), (ii.b), (ii.c) and omit the i in the subscripts to improve readability. Differently shaped/colored vertices correspond to different requests of K .



■ **Figure 8** Partial representation of the gadgets $D_{i,h}$ and $D_{l,r}$ with the thicker arcs being those added in Step (iii) of the reduction. The grayed path encodes one edge of the multicolored clique of G . Differently shaped/colored vertices correspond to different requests of K .

Note that, if we can guarantee that each $i \in [q]$ has exactly one $D_{i,j}$ unoccupied by the paths discussed in Step (ii), then the paths used to satisfy each $\delta_{i,\ell}$ pair must, necessarily, go through arcs of D that encode adjacency in G , and all such arcs must be incident to the same vertex of V_i . We now formalize our intuition in the lemmas required to prove our main result for this section.

► **Observation 14.** *Instance (D, K) of DIRECTED DISJOINT PATHS has $|K| = 6q + \binom{q}{2}$, D can be partitioned into $6q + 2\binom{q}{2}$ tournaments and has directed pathwidth equal to 2.*

Proof. The first property has already been proved during the construction. The second one follows almost immediately: each D_i has four non-trivial tournaments: $D_i^\alpha \cup \{g_{i,s}^2\}$, $D_i^a \cup \{g_{i,s}^1\}$, $D_i^\beta \cup \{g_{i,t}^1, g_{i,t}^2\}$, and D_i^b ; aside from these, D has $2\binom{q}{2}$ trivial tournaments of the form $\delta_{i,\ell,*}$ with $i < \ell$ plus $2q$ trivial tournaments with $i = \ell$.

The final property requires a bit more of care, but note that each x -path of D_i is a digraph of directed pathwidth 2: there is a path decomposition P_i^x of width 2 where each bag contains one arc of the x -path and consecutive bags have consecutive arcs. As such, we can compose a width-2 decomposition of D_i by ordering its vertices as $P_i = \langle g_{i,t}^1, g_{i,t}^2, P_i^\beta, P_i^b, P_i^a, P_i^\alpha, g_{i,s}^1, g_{i,s}^2 \rangle$. Finally, we order D as:

$$\begin{aligned} & \langle \delta_{i,\ell,t} \mid i \in [q], \ell \in [q] \setminus [i-1] \rangle, \\ & \langle P_q, \dots, P_1 \rangle, \\ & \langle \delta_{i,\ell,t} \mid i \in [q], \ell \in [q] \setminus [i-1] \rangle. \end{aligned}$$

Note that the P_i 's are ordered *decreasingly*, while the other two listed suborders can be arbitrarily internally ordered. ◀

We could modify our reduction to get a smaller number of cliques, but breaking the $\Theta(q^2)$ bound for either $|K|$ or the number of cliques seems very hard with this approach. As such, we do not further trim down either parameter, opting for the current version for ease of exposition.

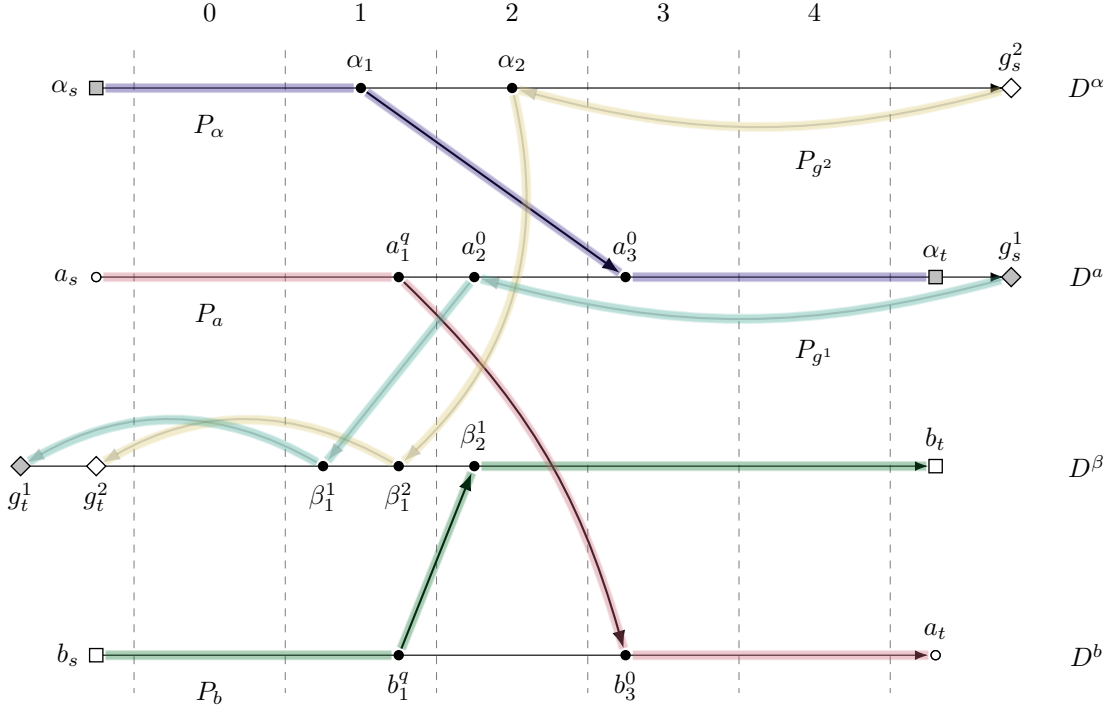
► **Lemma 15.** *If there is a solution Q to the MULTICOLORED CLIQUE instance (G, \mathcal{V}) , then the DIRECTED DISJOINT PATH instance (D, K) admits a solution \mathcal{P} .*

Proof. Let $u \in V_i \cap Q$. To satisfy the internal paths of D_i , denoted by \mathcal{I}_i , we add the following disjoint paths, which are illustrated in Figure 9:

$$\begin{aligned} (\alpha_{i,s}, \alpha_{i,t}) : & \quad \langle \alpha_{i,s}, \alpha_{i,0}, \dots, \alpha_{i,u-1}, a_{i,u+1}^0, \dots, a_{i,n+1}^q, \alpha_{i,t} \rangle \\ (a_{i,s}, a_{i,t}) : & \quad \langle a_{i,s}, a_{i,0}^0, \dots, a_{i,u-1}^q, b_{i,u+1}^0, \dots, b_{i,n+1}^q, a_{i,t} \rangle \\ (b_{i,s}, b_{i,t}) : & \quad \langle b_{i,s}, b_{i,0}^0, \dots, b_{i,u-1}^q, \beta_{i,u}^1, \dots, \beta_{i,n+1}^2, b_{i,t} \rangle \\ (g_{i,s}^2, g_{i,t}^2) : & \quad \langle g_{i,s}^2, \alpha_{i,u}, \beta_{i,u-1}^2, g_{i,t}^2 \rangle \\ (g_{i,s}^1, g_{i,t}^1) : & \quad \langle g_{i,s}^1, a_{i,u}^0, \beta_{i,u-1}^1, g_{i,t}^1 \rangle. \end{aligned}$$

To satisfy the requests involving the δ vertices, we add $\langle \delta_{i,i,s}, a_{i,u}^i, b_{i,u}^i, \delta_{i,i,t} \rangle$ to \mathcal{P} . Now, take $v \in V_\ell \cap Q$ with $i < \ell$. We add the path $P_{i,\ell}$, which is displayed in Figure 8:

$$P_{i,\ell} = \langle \delta_{i,\ell,s}, a_{i,u}^\ell, b_{i,u}^\ell, a_{\ell,v}^i, b_{\ell,v}^i, \delta_{i,\ell,t} \rangle.$$



■ **Figure 9** Paths used to satisfy the interior requests of a gadget D_i if $j = 2$; the i subscript is omitted for readability. Shaded contiguous paths represent the appropriate paths added to the solution of (D, K) to satisfy the long paths of D_i , while the gray arcs are used to satisfy its short paths. Differently shaped/colored vertices correspond to different requests of K .

This concludes the construction of \mathcal{P} . By definition, a path $P_{i,\ell}$ occupies vertices of $D_{i,u}$ if and only if $u \in Q \cap V_i$, while paths of \mathcal{I}_i only occupy vertices $a_{i,u}^0, b_{i,u}^0$, which are never used by $P_{i,\ell}$, so it follows that $P_{i,\ell}$ and \mathcal{I}_i are disjoint. Finally, $P_{i,\ell}$ and $P_{i,r}$ are also disjoint whenever $\ell \neq r$, so \mathcal{P} is indeed a collection of $|K|$ disjoint paths satisfying K . ◀

The converse direction, as usual, requires additional care in its proof. To this end, we first show that minimal solutions in (D, K) must adhere to a specific format, i.e., the jumps between the x -paths of each D_i happen *synchronously* as in Figure 9: all forward jumps leave a gap at precisely one index j of each D_i , and this happens due to how the backward jumps were defined.

► **Lemma 16.** *Let \mathcal{P} be a solution of (D, K) where every $P \in \mathcal{P}$ is induced (i.e., P is minimal), $D_i \in D$, $P_{i,i} \in \mathcal{P}$ be the path satisfying $(\delta_{i,i,s}, \delta_{i,i,t}) \in K$, and \mathcal{I}_i be the set of paths satisfying internal requests of D_i . It holds that (1) $V(\mathcal{I}_i) \subseteq D_i$ and there is some $u \in [n]$ such that: (2) all forward jumps performed by paths of \mathcal{I}_i originate in $D_{i,u-1}$, and (3) all backward jumps go from $D_{i,u} \setminus D_{i,u-1}^\beta$ to $D_{i,u-1}^\beta$. Moreover, (4) for every $v \in \{0, \dots, n+1\} \setminus \{u\}$, we have $D_{i,v}^a \cup D_{i,v}^b \subseteq V(\mathcal{I}_i)$ and $(D_{i,u}^a \cup D_{i,u}^b) \cap V(\mathcal{I}_i) = \{a_{i,u}^0, b_{i,u}^0\}$.*

Proof. To prove (1), observe that the vertices of $G \setminus V(K)$ that can be reached from a vertex of D_i are precisely those in D_ℓ for $\ell > i$; that is, paths of \mathcal{I}_i only use vertices of D_i .

Towards proving (2) and (3), take $P_{i,b}$ as the path satisfying request $(b_{i,s}, b_{i,t})$ and note that it *must* use a forward jumping arc $(b_{i,u-1}^q, \beta_{i,u}^1)$ for some $u \in [n]$, as these are the only arcs that connect the b - and β -paths of D_i and to reach $b_{i,t}$ we must use part of the β -path

of D_i . Moreover, this arc is unique, as D_i^β has *no* arc leaving it. Since the only arcs going from D_i^b to D_i are of this form, it follows that:

$$P_{i,b} = \langle b_{i,s}, b_{i,0}^0, \dots, b_{i,u-1}^q, \beta_{i,u}^1, \dots, \beta_{i,n+1}^2, b_{i,t} \rangle.$$

Define $P_{i,a}$ similarly, that is, the path satisfying request $(a_{i,s}, a_{i,t}) \in K$, and note that it must reach D_i^b at some point $b_{i,v}^x$ for some $v \geq u$ and, after that, it only occupies vertices of D_i^b . Consequently, the last vertex of D_i^a in $P_{i,a}$ belongs to the suffix of the a -path that starts at $a_{i,u-1}^q$ and its superscript is not 0, as these vertices have no arc to D_i^b . Moreover, $P_{i,a}$ contains the prefix of the a -path ending at $a_{i,u-1}^{q-1}$: these vertices' only neighbors are either in D_i^β , which is a dead-end for $P_{i,a}$, or have already been used by $P_{i,b}$. The sole exception to this analysis is vertex (if it exists) $a_{i,u-2}^q$, which has $b_{i,u}^0$ not yet accounted for; but if this jump is performed, then $D_i^b \subset P_{i,a} \cup P_{i,b}$ and becomes impossible to satisfy the request $(\delta_{i,i,s}, \delta_{i,i,t}) \in K$. If the last vertex of D_i^a in $P_{i,a}$ was not $a_{i,u-1}^q$ we would have $a_{i,u}^0 \in P_{i,a}$, but then observe that $(g_{i,s}^1, g_{i,t}^1) \in K$ would be unsatisfiable: the only neighbors of $a_{i,v}^0$ with $v > u$, which are the only neighbors of $g_{i,s}^1$ not in $P_{i,a}$, would already be occupied by $P_{i,b}$. As such, we conclude that:

$$P_{i,a} = \langle a_{i,s}, a_{i,0}^0, \dots, a_{i,u-1}^q, b_{i,u+1}^0, \dots, b_{i,n+1}^q, a_{i,t} \rangle.$$

Let us now look at the paths P_{i,g^x} that satisfy the requests $(g_{i,s}^x, g_{i,t}^x)$. The above also implies that $P_{i,g^1} = \langle g_{i,s}^1, a_{i,u}^0, \beta_{i,u-1}^1, g_{i,t}^1 \rangle$. Now, observe that the first arc of P_{i,g^2} must have $\alpha_{i,v}$ as an endpoint and $v \leq u$; since P_{i,g^2} is minimal, no other α_i vertex is in it and $v \not\geq u$ as $D_{i,w}^\beta \subset P_{i,b}$ whenever $w > u$. In fact, it must be the case that $v = u$: the $P_{i,\alpha}$, which satisfies the request $(\alpha_{i,s}, \alpha_{i,t})$, cannot use vertices of D_i^β and it cannot perform its unique jump to D_i^a before $\alpha_{i,u-1}$, or it would clash with $P_{i,a}$. Together, these imply that:

$$\begin{aligned} P_{i,\alpha} &= \langle \alpha_{i,s}, \alpha_{i,0}, \dots, \alpha_{i,u-1}, a_{i,u+1}^0, \dots, a_{i,n+1}^q, \alpha_{i,t} \rangle \\ P_{i,g^2} &= \langle g_{i,s}^2, \alpha_{i,u}, \beta_{i,u-1}^2, g_{i,t}^2 \rangle. \end{aligned}$$

Consequently, (2) and (3) hold. Finally, (4) follows immediately from the proofs of (1)-(3). \blacktriangleleft

Property (4) of [Lemma 16](#) is the key ingredient of our proof and the main complication to generalize Slivkins' result. In particular, we need to fully use D_i^a and D_i^b to avoid cheating by the δ paths: without this guarantee, we could have a δ path hitting an a vertex at some index u , going to an a vertex of index v using an arc added to force the appearance of a tournament, moving to its matched b vertex and then moving on to any D_j , completely breaking down the desired behavior. With these bizarre paths safely ruled out, proving the converse becomes quite direct, as we show in the following lemma.

► **Lemma 17.** *If (D, K) admits a solution \mathcal{P} , then (G, \mathcal{V}) contains a multicolored clique Q of appropriate size.*

Proof. Suppose that \mathcal{P} is minimal, i.e., none of its paths has a shortcut. By [Lemma 16](#), for each $i \in [q]$ there is exactly one $u \in [n]$ that is not completely occupied by $V(\mathcal{I}_i)$. We claim that the collection of vertices $Q \subseteq V(G)$ corresponding to these indices forms a clique in G . It suffices to show that $u, v \in Q$ implies $uv \in E(G)$. First, note that every $D_{i,w}^a$ for

$i \in [q], w \in Q$ must be occupied by a path that satisfies a request between δ vertices, and each one is occupied by a different path; this follows directly by induction on i , with the base case $i = 1$ being straightforward. Suppose that $u \in V_i, v \in V_\ell$ and $i < \ell$ and let $P_{i,\ell}$ be the path satisfying $(\delta_{i,\ell,s}, \delta_{i,\ell,t}) \in K$. Again by Lemma 16, $a_{i,u}^\ell$ is the unique out-neighbor of $\delta_{i,\ell,s}$ not occupied by another path, so $(\delta_{i,\ell,s}, a_{i,u}^\ell) \in P_{i,\ell}$. By a similar argument, $(a_{i,u}^\ell, b_{i,u}^\ell) \in P_{i,\ell}$. Since $P_{i,\ell}$ exists and $b_{i,u}^\ell$ is not its terminal, the latter must have exactly one out-neighbor in the former. By our construction, this vertex must be in D_ℓ^a and, by Lemma 16, this is precisely $a_{\ell,v}^i$. In turn, this arc only exists if $uv \in E(G)$, concluding the proof. ◀

► **Theorem 4.** *The k -DDP problem on C_h is $W[1]$ -hard when parameterized by $k + h$, even if restricted to input digraphs of directed pathwidth two.*

Proof. The proof follows directly from Observation 14 and Lemmas 15 and 17. ◀

4.3.1 Into the congestionverse

We employ a trick very similar to the one used in Subsubsection 4.2.3. Essentially, we introduce a path that is *almost* Hamiltonian, missing only the δ vertices of Theorem 4. It is also possible to include them in this new path, but for simplicity's sake we do not. Formally, we proceed as follows: let (D', K') be the instance of DIRECTED DISJOINT PATHS built in the proof of Theorem 4 and $c > 1$ be an integer; to obtain (D, K, c) , we add $c - 1$ requests of the form (z_s, z_t) to K' and the corresponding vertices to D' . Now, add the arcs $(z_s, g_{q,t}^1)$ and $(g_{1,s}^2, z_t)$ to D . Then, for each $i \in [q]$, we add the arcs $(b_{i,t}, b_{i,s}), (a_{i,t}, a_{i,s}), (g_{i,s}^1, \alpha_{i,s})$ and, if $i > 1$, $(g_{i,s}^2, g_{i-1,t}^1)$. This completes the construction of (D, K) ; note that $|K| = |K'| + (c - 1)$ and that the directed pathwidth of D remains 2.

► **Observation 18.** *Let \mathcal{P} be a solution to (D, K) . There exists a unique path P_z that can satisfy (z_s, z_t) and, for every $i \in [q]$, this path is obtained by concatenating the β -, b -, a - and α -paths, in this order.*

Proof. Our instance (D', K') in Theorem 4 was carefully constructed so that D^β was a path with no outgoing arc; we used this extensively in the proof of Lemma 16. This is no longer true for (D, K) , but note that the *unique* outgoing arc is from $b_{q,t}$; that is, in order for a path to leave D_q^β it must necessarily pass through this arc, so it follows that D_q^β belongs to every path from z_s to z_t . Consequently, the path now finds itself at $b_{q,s}$ and, aside from $(a_{q,t}, a_{q,s})$, the only arcs that leave D_q^b go to D_q^β , but the latter has already been used by the path we are constructing, implying that P_z must also traverse the entire b -path of D_q and then leave through $(a_{q,t}, a_{q,s})$. This happens again with the a - and α -paths of D_q , with P_z using the arc $(g_{q,s}^2, s, g_{q-1,t}^1)$ to arrive at D_{q-1} ; note that every vertex reachable by D_{q-1} that is not in D_{q-1} has already been visited by P_z , so the argument extends all the way to D_1 , when P_z is finally able to reach z_t . ◀

As in the proof of Theorem 12, we are now free to choose c to get whichever fraction of K we desire, and so we obtain the following theorem.

► **Theorem 19.** *(k, c) -DDP jointly parameterized by the number of requests $|K|$ and the minimum clique cover of the host graph is $W[1]$ -hard even if the input instance has directed pathwidth 2 and the congestion c is a fraction of, but not equal to, $|K|$.*

5 Algorithmic results

In this section we prove our main algorithmic result and discuss its applications.

► **Theorem 20.** *For all integers $k \geq 1$ and $h \geq 0$, there is a function $f(k, h)$ such that every instance of (k, c) -DDP with $2c > k$ on an h -semicomplete digraph D containing a $f(k, h)$ -triple has an irrelevant vertex which can be found in $\mathcal{O}(f^2(k) \cdot n^2)$ time. In particular, $f(k, 0) = \mathcal{O}(2^{k \cdot \log k})$.*

Fomin and Pilipczuk [23, Theorem 6.3] showed that DIRECTED EDGE-DISJOINT PATHS is FPT on tournaments parameterized by the number of paths. They solve the problem with a win-win approach. First they show that the problem is FPT parameterized by the directed pathwidth $\text{dpw}(T)$ of the input semicomplete digraph T plus the number k of paths. Then, through a series of results and constructions, they show that there is a computable function $g(k)$ such that every semicomplete digraph T with $\text{dpw}(T) \geq g(k)$ contains a k -triple, and that every such triple contains a vertex v that is irrelevant for the given instance. This second proof essentially shows that only the instances with $\text{dpw}(T)$ bounded by some function of k need to be solved.

Although they leave open whether DIRECTED DISJOINT PATHS is FPT on semicomplete digraphs, they mention that an FPT dynamic programming algorithm, parameterized by the number of paths, can be done for DIRECTED DISJOINT PATHS to obtain a result analogous to [23, Theorem 6.3].

Another approach to obtain such an FPT algorithm relies on *directed cliquewidth*, which is bounded from above by $\text{dpw}(T) + 2$ on every semicomplete digraph T [23, Lemma 2.14]. Then one can apply [23, Theorem 2.16] which states that, given an MSO_1 formula ϕ and a semicomplete digraph T , there is an algorithm which checks whether ϕ is satisfied by T in FPT time parameterized by the order of ϕ and $\text{dpw}(T)$. This implies the existence of an FPT algorithm on semicomplete digraphs for DIRECTED DISJOINT PATHS parameterized by the number of paths, since this problem can be modeled by MSO_1 (see, for instance, [28, Proposition 4.7]). In any case, the following is obtained.

► **Proposition 21** (Fomin and Pilipczuk [23]). *DIRECTED DISJOINT PATHS on semicomplete digraphs is FPT parameterized by the number of paths and the directed pathwidth of the input semicomplete digraph.*

Applying classical techniques, one can easily show that Proposition 21 also implies an FPT algorithm for (k, c) -DDP. Indeed, given an instance of this problem on a semicomplete digraph T , it suffices to add copies v_2, \dots, v_c of each $v \in V(T)$ with the same in- and out-neighborhood as v . Then, we add arcs among vertices in $\{v, v_2, \dots, v_c\}$ in order to ensure that this set induces an acyclic tournament. This procedure easily implies that the directed pathwidth of T increases by at most a factor of c . Hence we can apply Proposition 21 to solve k -DDP in the newly constructed digraph and transport any solution, whether positive or negative, to the original instance in T . This implies that (k, c) -DDP is also FPT parameterized by k and the directed pathwidth of the input digraph.

When applying their irrelevant vertex rule, Fomin and Pilipczuk [23] use a series of results to construct in FPT time a triple of large order on semicomplete digraphs of sufficiently large directed pathwidth. Namely, given a semicomplete digraph T and an integer k , they first apply [23, Theorem 4.12] to either produce a directed path decomposition of width bounded by some computable function $g(k)$ or find one of the two certificates that $\text{dpw}(T) > k$: a *degree tangle* or a *matching tangle*. Then, they show how to produce a *short jungle* from any

of those two objects in [23, Lemmas 3.9 and 3.12]. We remark that all these constructions run in polynomial time. They argue that although the proof of how to extract triples from short jungles by Fradkin and Seymour [26] is not explicitly algorithmic, it is easy to extract an algorithm from it. In any case, the proof of [23, Theorem 9.9] shows how to circumvent the lack of an explicit algorithmic result to obtain triples from jungles, essentially obtaining the following result one way or another when applying the other results mentioned in this paragraph.

► **Proposition 22** (Fomin and Pilipczuk [23]). *Let T be a semicomplete digraph. There is a computable function $g(t)$ and an FPT algorithm parameterized by t that either outputs a directed path decomposition of T with width at most $g(t)$ or finds a t -triple in T .*

In light of Propositions 21 and 22, it is now easy to apply Theorem 20 to obtain an FPT algorithm for (k, c) -DDP when $2c > k$. We restate Theorem 2 for convenience.

► **Theorem 2.** *(k, c) -DDP on semicomplete digraphs is FPT parameterized by k restricted to instances satisfying $c > k/2$.*

Proof. Let \mathcal{I} be an instance of (k, c) -DDP on a semicomplete digraph with $2c > k$. Let $f(k, h)$ be the function as in the statement of Theorem 20, and $f(k) = f(k, 0)$. Applying Proposition 22 with input T and $t = f(k)$, we either receive a directed path decomposition of T with width at most $g(f(k))$ or an $f(k)$ -triple in T . In the first case, we solve the instance applying Proposition 21 (we remind the reader of the discussion immediately after this proposition). In the second case, we apply Theorem 20 to find a vertex $v \in V(T)$ that can be deleted without changing the answer to \mathcal{I} .

After deleting an irrelevant vertex, we apply Proposition 22 again. Each application of this proposition that yields the second possible output results in the deletion of one vertex from T . Thus, after at most $|V(T)|$ iterations the first output is guaranteed to occur, and we solve the problem applying Proposition 21. ◀

5.1 Irrelevant vertices in k -triples

This section is dedicated to the proof of Theorem 20.

We first prove the version of Theorem 20 restricted to semicomplete digraphs, and at the very end of this section we explain which are the (easy) changes to be made to the proof so that it holds for any $h \geq 1$. We abbreviate $f(k, 0)$ by $f(k)$ and assume that an $f(k)$ -triple is given. The choice of $f(k)$ is discussed later. Let $\mathcal{I} = (T, K, c)$ be an instance of (k, c) -DDP on a semicomplete digraph T with $2c > |K| = k$. For the remainder of this section, as well as in Subsection 5.2, Subsection 5.3, and Subsection 5.4, we assume that \mathcal{K} is an $f(k)$ -triple (A, B, C) of T and that \mathcal{P} is a set of paths satisfying the following property.

► **Property 23.** \mathcal{P} is a solution $\{P_1, \dots, P_k\}$ for \mathcal{I} minimizing $\sum_{i \in [k]} |V(P_i)|$.

Since congestion is allowed, any vertex is counted in the summation as many times as it is used by a path. This per-path measure is necessary as we often shortcut (we formally define shortcuts later in this section) a path $P \in \mathcal{P}$ through a vertex that is used by other paths in the collection \mathcal{P} .

To avoid some technicalities, here we make two assumptions about \mathcal{K} . First, by discarding at most $2k$ vertices of each set A, B , and C , we assume that the terminals of \mathcal{I} do not appear in $V(\mathcal{K})$. Second, by deleting parallel arcs of T if necessary, we assume that there are no arcs in $E(\mathcal{K})$ from C to B and no arcs from B to A . The removal of those arcs poses no issue for the irrelevant vertex argument since any irrelevant vertex of $T' \subseteq T$ is also irrelevant in T .

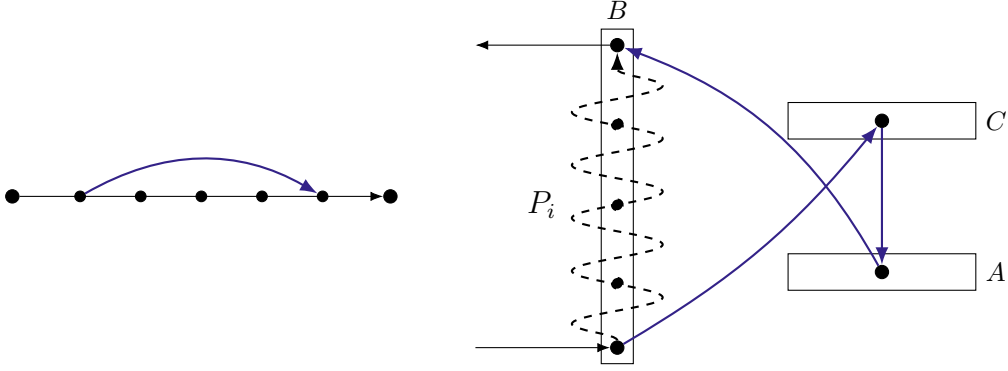
The set B plays a distinguished role in this proof since it concentrates the heads of all arcs of \mathcal{K} between A and B and the tails of all arcs of \mathcal{K} between B and C . The goal is to show that there is room to reroute paths in \mathcal{K} in order to show that if \mathcal{I} is a **yes**-instance, then there is a vertex $b \in B$ and a solution \mathcal{Q} for \mathcal{I} such that every $Q \in \mathcal{Q}$ accessing b does so from a vertex in A and to a vertex in C .

We follow the blueprint of the proof for DIRECTED EDGE DISJOINT PATHS on semicomplete digraphs by Fomin and Pilipczuk [23]. A fundamental property in their case is that any vertex with out-degree (resp. in-degree) at least $k + 1$ in the given triple has at least one arc leaving (resp. entering) it that is not used by any solution \mathcal{Q} minimizing the sum of the lengths of its paths. This property allows them to prove that every such solution cannot use more than two arcs of the matching from C to A . This, in turn, is used to show that every path in \mathcal{Q} uses at most $2k + 4$ vertices from A and from C , and at most $4k$ vertices of B . The hard part of their proof is about how to apply these results to prove that an irrelevant vertex is guaranteed to exist in a sufficiently large triple, and how to find it in polynomial time.

In our case, the existence of a vertex $v \in V(\mathcal{K})$ with large out-degree or in-degree in the triple does *not* guarantee that we can find a vertex in $N^+(v) \cup N^-(v)$ that is not used by a path of \mathcal{P} . In fact, in many places of our proof we do not guarantee that at all, since we produce shortcuts for paths in \mathcal{P} through vertices that have been used by other paths of \mathcal{P} while but *at most* $c - 1$ of them, hence being careful not to exceed the allowed congestion on each vertex. The fundamental property that we use comes from the assumption that $2c > k$: if u, v are both used by c paths of \mathcal{P} , then by the pigeonhole principle there is a path $P \in \mathcal{P}$ that uses *both* u and v . The need to rely on this property is justified by the counterexample provided in Subsection 3.3, which shows that no irrelevant vertex is guaranteed to exist in arbitrarily large triples whenever $2c \leq k$, and makes our analysis significantly harder than the one in [23]. For example, in order to show that a bounded number of vertices of A and C are used by paths of \mathcal{P} (Lemma 26), we first need to show that some vertices of B can be used for shortcuts (Lemma 24) and these two proofs are already much harder than their counterparts in [23].

In addition to the definitions introduced in the beginning of this section, we adopt the following notation. We denote by \mathbf{M} the arcs of the matching from C to A and, as discussed right below Definition 5, we may also refer to \mathbf{M} as a bijective mapping from C to A according to the arcs of the matching defining \mathcal{K} . In addition, for $C' \subseteq C$ (resp. $A' \subseteq A$) we call the set $\mathbf{M}(C') = \{\mathbf{M}(v) \mid v \in C'\}$ (resp. $\mathbf{M}^{-1}(A') = \{\mathbf{M}^{-1}(u) \mid u \in A'\}$) the *mirror* of C' (resp. of A') in \mathcal{K} .

For $P_i \in \mathcal{P}$ we denote by \prec_i the order in which $V(P_i)$ appears in P_i . A *shortcut* for P_i is a walk R such that there is some subpath $R' \subseteq P_i$ where $|V(R)| < |V(R')|$ and R starts and ends in the same vertices as R' . See Figure 10 for examples of shortcuts. We may refer to a shortcut R as the sequence $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_m$ of the vertices of R as they appear in the walk. The existence of a shortcut for any path in \mathcal{P} respecting the congestion measure contradicts Property 23, and the goal of the first part of the proof of Theorem 20 is to exploit this fact to prove useful properties on how the paths of \mathcal{P} can intersect an $f(k, h)$ -triple. To avoid repetition, we refrain from stating that these shortcuts lead to contradictions in the proofs of this section. Note that simply replacing R' with R may not lead to a minimal solution: there could be forward arcs that further shorten the resulting object, or cycles could be introduced, depending on the visiting order of the involved vertices. For example, in Figure 10, it could be the case that the vertex of C appeared in the depicted path *after* the vertex of A , but with the addition of the blue segment we now have a cycle.



■ **Figure 10** Examples of shortcuts. In both cases, the blue path denotes a shortcut. On the right, we give an illustration of a typical shortcut, internal to a k -triple (A, B, C) , that is often built in the proof of [Theorem 20](#). The dashed subpath of P_i is exchanged by the blue path in the figure.

For every $v \in V(\mathcal{K})$, we assign to v a list of indices $\mathbf{L}(v) \subseteq 2^{[k]}$ representing which paths of \mathcal{P} are using v . Thus $|\mathbf{L}(v)| \leq c$ holds for any vertex v since \mathcal{P} is a solution for \mathcal{I} . For $i \in [k]$, we say that v is i -free if $i \in \mathbf{L}(v)$ or if $|\mathbf{L}(v)| \leq c - 1$, and that v is i -saturated otherwise. Intuitively, an i -free vertex can be used to construct shortcuts for path P_i since either P_i is already using v and thus the route may be replaced with a shorter one without increasing the congestion of v , or the congestion of v is not yet saturated and can be increased by one to generate a better solution. We denote by \mathcal{L} the set of all possible lists of indices that can be assigned to a vertex by a solution. More precisely, $\mathcal{L} = \{L \subseteq 2^{[k]} \mid |L| \leq c\}$. For $L \in \mathcal{L}$ we denote by $V(L)$ the set $\{v \in V(\mathcal{K}) \mid \mathbf{L}(v) = L\}$ of vertices of \mathcal{K} which are assigned indices in L . Finally, for $u \in C$ and $v = \mathbf{M}(u)$ we say that $\{u, v\}$ is an i -free pair if both u and v are i -free.

5.2 Freeing B

In this section, we show that only a bounded number of vertices $b \in B$ have $|\mathbf{L}(b)| = c$. We begin with the following warm up lemma.

► **Lemma 24.** *For all $P_i \in \mathcal{P}$, if $|V(P_i) \cap B| \geq 5$ then there are no i -free pairs.*

Proof. By contradiction, assume that there is $P_i \in \mathcal{P}$ such that $|V(P_i) \cap B| \geq 5$ and there is an i -free pair u, v with $u \in C$ and $v = \mathbf{M}(u)$. Let $\langle b_1, \dots, b_5 \rangle$ be the first five vertices of P_i that appear in B . If none of $u, v \notin V(P_i)$ then we can shortcut the subpath of P_i from b_1 to b_5 by following the arcs $b_1 \rightarrow u \rightarrow v \rightarrow b_5$. Thus at least one of u, v appears in P_i .

If both $u, v \in V(P_i)$ then $b_4 \prec_i v$, since otherwise we can shortcut using $v \rightarrow b_5$, and $u \prec_i b_2$, since otherwise $b_1 \rightarrow u$ is a shortcut. As such, we have that $u \prec_i b_2 \prec_i b_4 \prec_i v$, and we shortcut through $b_1 \rightarrow u \rightarrow v \rightarrow b_5$. The cases when only one of u, v is not in P_i follow similarly. ◀

► **Lemma 25.** *For all $L \in \mathcal{L}$ with $|L| = c$, it holds that $|V(L) \cap B| \leq 4$.*

Proof. By contradiction, assume that there is $L \in \mathcal{L}$ such that $|L| = c$ and $|V(L) \cap B| \geq 5$. For all $i \in L$, let $A_i = V(P_i) \cap A$ and $C_i = \mathbf{M}^{-1}(A_i)$.

If all $i \in L$ satisfy $|A_i| \leq 4c$, then $|V(L) \cap A| \leq 4c^2$ and thus, since $|L| = c$, the set $A' = A \setminus V(L)$ has size at least $f(k) - 4c^2$ and all its vertices are used by at most $c - 1$ paths of \mathcal{P} since $2c > k$. By definition, all vertices in A' are j -free for all $j \in [k]$. If there is

$u \in M^{-1}(A')$ with $|L(u)| = c$, then it holds that there is some $j \in L \cap L(u)$ and, consequently, that $\{u, M(u)\}$ is a j -free pair. Otherwise, $|L(u)| \leq c - 1$ for all $u \in M^{-1}(A')$. Both cases contradict [Lemma 24](#) and the result follows.

Assume now that there is an $i \in L$ such that $|A_i| \geq 4c + 1$ and notice that every $u \in C_i$ has $|L(u)| = c$, as otherwise $\{u, M(u)\}$ would be an i -free pair, contradicting [Lemma 24](#) as $|V(P_i) \cap B| \geq |V(L) \cap B| \geq 5$. The goal now is to group all pairs of the form $\{u, M(u)\}$ with $u \in C_i$ according to which indices appear in $L(u) \cap L(M(u))$. The size of $|A_i|$ guarantees that at least one index ℓ appears in the lists of the endpoints of many of these pairs and we use it to shortcut two paths. First, we shortcut P_ℓ through some $b \in V(L) \cap B$. By doing this, we no longer have a solution for our instance since a vertex of this set is used by $c + 1$ paths. However, the shortcut for P_ℓ frees some vertices of C , which are now used by at most $c - 1$ paths. Thus we can shortcut P_i in a way that avoids b to generate a new solution contradicting [Property 23](#).

Formally, we proceed as follows. For $j \in [k] \setminus L$, let X_j be the set of pairs $\{u, M(u)\}$ such that $j \in L(u) \cap L(M(u))$ with $u \in C_i$. Notice that these sets are not necessarily pairwise disjoint and that the exclusion of L is natural as the occurrence of any of its indices in $L(u) \cap L(M(u))$ would contradict [Lemma 24](#). In addition, the pigeonhole principle implies that at least one set X_ℓ satisfies $|X_\ell| \geq 5$ as $|A_i| \geq 4c + 1$.

Let A' be the subset of vertices of A appearing in pairs of X_ℓ and $C' = M^{-1}(A')$, and let b_1, \dots, b_q be the first $q \geq 5$ vertices of $V(L) \cap B$ ordered as they appear in P_i . We now consider two cases.

1. Assume first that P_ℓ uses a vertex of A' before any vertex of C' . Denote by v_1 the first vertex of A' used by P_ℓ , and by u the last vertex of C' used by P_ℓ . We construct the path P'_ℓ by shortcutting P_ℓ through $v_1 \rightarrow b_3 \rightarrow u$. This indeed forms a shortcut since at least four vertices used by P_ℓ in C' are skipped by P'_ℓ . The collection $\mathcal{P}' = (\mathcal{P} \setminus \{P_\ell\}) \cup \{P'_\ell\}$ improves on the summation in [Property 23](#) with relation to \mathcal{P} , but b_3 is now used by $c + 1$ paths and thus \mathcal{P}' is not a solution for the instance \mathcal{I} . The choice of P'_ℓ ensures that at least three vertices of $C' \setminus \{M^{-1}(v_1)\}$ are *not* used by it, and therefore we can choose one of those vertices, say u' , to shortcut P_i through $b_1 \rightarrow u' \rightarrow M(u') \rightarrow b_q$; by using this new path instead of P_i in \mathcal{P}' , we remove i from the list of b_3 and no other vertex exceeds the allowed congestion, thus contradicting [Property 23](#). See [Figure 11](#) for an example of the shortcuts that are built in the analysis of this case.

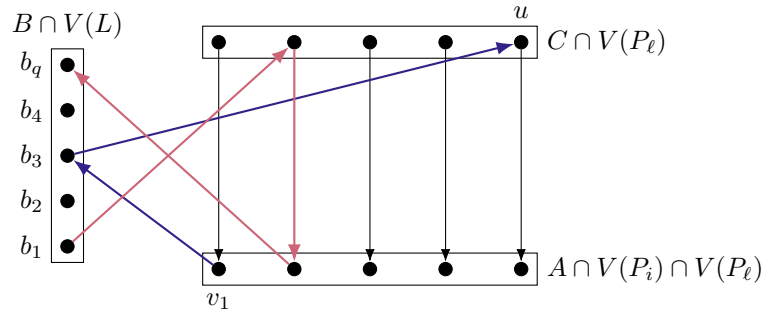


Figure 11 Shortcuts built in the first case of the proof of [Lemma 25](#). The blue path starting in v_1 is the shortcut for P_ℓ , forming P'_ℓ . The red path starting in b_1 is the shortcut for P_i . The remaining arcs of \mathcal{K} are omitted.

2. Assume now that P_ℓ uses a vertex of C' before any vertex of A' . Denote by u_1 and w the first and last vertices of C' used by P_ℓ , respectively. Similarly to the last case, we form

P'_ℓ by shortcutting P_ℓ through $u_1 \rightarrow M(u_1) \rightarrow b_3 \rightarrow w$, thus increasing the congestion on the vertex b_3 to $c + 1$. Again, there are at least three vertices of C' not used by P'_ℓ and we use one of those, say u' , to shortcut P_i through $b_1 \rightarrow u' \rightarrow M(u') \rightarrow b_q$ in order to avoid b_3 . This contradicts [Property 23](#) and the result follows. Notice that the fact that $|A'| \geq 5$ is needed only in this case due to the size of the shortcut for P_ℓ . \blacktriangleleft

[Lemma 25](#) implies that at least $f(k) - 4\binom{k}{c}$ vertices of B are used by at most $c - 1$ paths of \mathcal{P} . Thus, all such vertices are i -free for any $i \in [k]$ and can be used to build shortcuts for any path in \mathcal{P} .

5.3 Freeing A and C

We now show that only a bounded number of vertices of A and C are used by paths of \mathcal{P} .

► **Lemma 26.** *If there is $b \in B$ such that $|L(b)| \leq c - 1$, then for all $P_i \in \mathcal{P}$ it holds that*

1. $|V(P_i) \cap A| \leq 8c + 4$; and
2. $|V(P_i) \cap C| \leq 8c + 4$.

Proof. Let $b \in B$ with $|L(b)| \leq c - 1$. To prove 1, by contradiction let P_i be a path of \mathcal{P} using $q \geq 8c + 5$ vertices of A . Let $v_1 \prec_i v_2 \prec_i \dots \prec_i v_q$ be the vertices in $V(P_i) \cap A$ ordered as they appear in P_i . If $M^{-1}(v_j)$ is i -free for some $j \geq 5$, then we can shortcut P_i through $v_1 \rightarrow b \rightarrow M^{-1}(v_j) \rightarrow v_j$ to build a new solution. Thus, we now assume that every vertex in $C' = \{M^{-1}(v_j) \mid j \geq 5\}$ is i -saturated.

Let $A' = \{v \in M(C') \mid |L(v)| \leq c - 1\}$ and, towards a contradiction, assume that $|A'| \geq 4c + 1$. Then there is a $P_j \in \mathcal{P}$ using at least five vertices of $M^{-1}(A')$. If $u, u' \in M^{-1}(A')$ are the first and last vertices used by P_j in A' , respectively, then we can shortcut P_j through $u \rightarrow M(u) \rightarrow b \rightarrow u'$, again contradicting [Property 23](#). We conclude that $|A'| \leq 4c$.

Now, let $A'' = (V(P_i) \cap A) \setminus (A' \cup \{v_1, v_2, v_3, v_4\})$. Thus $|A''| \geq 4c + 1$ and every $w \in A'' \cup M^{-1}(A'')$ has $|L(w)| = c$. The remaining part of this proof follows similarly to what is done in the proof of [Lemma 25](#), but only one shortcut is needed since $|L(b)| \leq c - 1$. The size of A'' implies that there is some $\ell \in [k]$ and at least five vertices $v'_1, \dots, v'_5 \in A''$ such that $\ell \in L(M^{-1}(v'_j)) \cap L(v'_j)$ for $j \in [5]$. If P_ℓ uses a vertices of A'' before any vertex of its mirror and v'_1 and v'_5 are the first and last of the vertices used by P_ℓ in A'' , respectively, then we can shortcut this path through $v'_1 \rightarrow b \rightarrow M^{-1}(v'_5) \rightarrow v'_5$. The case when P_ℓ uses a vertex of $M^{-1}(A'')$ before any vertex of A'' is analogous and item 1 follows.

The proof of item 2 is mostly the same as in the first item, but with one small difference. By contradiction, let P_i be a path of \mathcal{P} using $q \geq 8c + 5$ vertices of C . Since all arcs between B and C are oriented from B to C , in this first step we need to exclude the four last vertices of P_i in C to say that the mirror of the remaining vertices of C contains only vertices which are used by c paths of \mathcal{P} . Let $u_1 \prec_i u_2 \prec_i \dots \prec_i u_q$ be the vertices in $V(P_i) \cap C$ ordered as they appear in P_i . If $M(u_j)$ is i -free for some $j \leq q - 4$, then we can shortcut P_i through $u_j \rightarrow M(u_j) \rightarrow b \rightarrow u_q$ to create a new solution. Thus, every vertex in $A' = \{M(u_j) \mid j \leq q - 4\}$ is i -saturated. The remaining part of the proof follows analogously to the proof of item 1 and the result follows. \blacktriangleleft

5.4 Finding the irrelevant vertex in polynomial time

Applying [Lemma 26](#) we show an improved version of [Lemma 25](#) which is needed in our proof.

► **Corollary 27.** *For all $i \in [k]$, $|V(P_i) \cap B| \leq 4$.*

Proof. Assume that there is a P_i using $z \geq 5$ vertices of B . Let b_1 and b_z be the first and last of those vertices under \prec_i , respectively.

Applying both items of [Lemma 26](#), we conclude that there is an i -free pair $\{u, M(u)\}$ with $u \in C$ after eliminating at most $2k(8c + 4)$ pairs $\{u', M(u')\}$ which can be intersected by $V(P_i)$. Thus we can shortcut P_i through $b_1 \rightarrow u \rightarrow M(u) \rightarrow b_z$ and the result follows. \blacktriangleleft

We are now ready to prove [Theorem 20](#) restricted to semicomplete digraphs. The goal is to give a polynomial-time algorithm that finds a vertex $b \in B$ such that if there is a solution for \mathcal{I} , then there is another solution whose paths all avoid b .

We follow the blueprint of the proof by Fomin and Pilipczuk [24, Theorem 9.1]. The goal is to first find a large set $X \subseteq B$ such that every path of a solution entering some $b' \in X$ from a vertex not in A can be rerouted to access b from a vertex in A . Then, we must show that there is a $b \in X$ such that every path of the solution leaving b to a vertex not in C can be rerouted to leave b to a vertex in C . As these rerouting steps do not use too many extra vertices in the triple, by [Corollary 27](#), and our hypothesis that the triple is large, we can then argue that b can be replaced with another $b^* \in B$, that is unused by the solution, and thus b can be safely removed from the graph.

Thus, at some point in the proof, we need to analyze how a path of a solution for \mathcal{I} can enter and leave B , and show how to reroute paths not respecting the desired behavior using vertices *outside* of the triple. In Fomin and Pilipczuk's proof, the fact that they consider collisions in arcs plays a major role here in this step. Informally, in a particular case of their analysis and using our notation, if a path P_i accesses $b' \in B$ through a vertex $v \in V(T) \setminus A$ and v has “sufficiently many” out-neighbors in $V(T) \setminus A$ which are in-neighbors of distinct vertices of B , then at least one of those out-neighbors, say v' , can be used to reroute P_i through $v \rightarrow v' \rightarrow u \rightarrow M(u) \rightarrow b$. This holds due to three main properties: (i) the fact that at most k arcs leaving a vertex are used by any solution, and thus any vertex of out-degree at least $k + 1$ has one arc leaving it that is free, (ii) their versions of [Lemmas 25](#) and [26](#), and (iii) a clever analysis on the behavior of the arcs between in-neighbors of vertices in B outside of A .

In our case, an analogous of the first of those three points requires more work since, a priori, it may seem that we have little control on how a solution intersects the in-neighborhood of B outside of A . We prove that this is not the case by applying another shortcutting argument which allows us to bound how many vertices a path $P_i \in \mathcal{P}$ can use in a particular subset of in-neighbors of B . In addition, after the first rerouting round, we can no longer rely on [Property 23](#), as we do increase the length of each path by an $\mathcal{O}(1)$ factor. Thus a small trick is needed to apply a similar shortcutting argument as the one used in the in-neighbors of B , this time to the out-neighbors of B ; this happens when we want to find a vertex of B that, when used by a path, is always followed by a vertex of C .

Proof of [Theorem 20](#) with $h = 0$. We remind the reader of the assumptions made in the beginning of [Subsection 5.1](#). Let:

$$\begin{aligned} d_2(k) &= 8k(4k + 1) + 8k + c, \\ m_2(k) &= 8k + c, \\ x(k) &= 2(d_2(k) \cdot m_2(k)), \\ d_1(k) &= 7k(4k + 1) + 8k + x(k), \text{ and} \\ m_1(k) &= 8k + x(k). \end{aligned}$$

Note that $x(k) = \mathcal{O}(k^3)$ is the asymptotically greatest among these five functions. However, in order to apply [Lemma 25](#), and consequently obtain [Lemma 26](#) and finally [Corollary 27](#), we need to start with $f(k) = \mathcal{O}(2^{k \log k})$. This is the only point of the proof where an exponential dependency on k is requested. [Corollary 27](#) hints that there might be a way to push this bound to a polynomial in k , but we do not see how to circumvent the $\binom{k}{c}$ blowup originating from [Lemma 25](#).

Since the size of B has the largest impact in the requested order of \mathcal{K} , in this proof we refrain from repeatedly proving that i -free pairs $u, \mathbf{M}(u)$ with $u \in C$ can be found when they are needed. By [Lemma 26](#) only $\mathcal{O}(k^2)$ such pairs are initially used by \mathcal{P} and each rerouting done below uses at most one extra pair.

First round of reroutings. The goal is to first find a set $X \subseteq B$ of size at least $x(k)$ such that if there is a solution for \mathcal{I} then there is another solution $\mathcal{Q} = \{Q_1, \dots, Q_k\}$ where:

- (i) for all $b \in X$, if b is accessed by some $v \in V(Q_i)$ with $i \in [k]$ then $v \in A$,
- (ii) at most $4k$ new vertices of B and $4k$ new pairs $\{u, \mathbf{M}(u)\}$ with $u \in C$ are used by \mathcal{Q} in comparison with \mathcal{P} , and
- (iii) each $Q_i \in \mathcal{Q}$ uses at most one new vertex of $V(T) \setminus V(\mathcal{K})$ in comparison with \mathcal{P} .

For every $b \in B$, let

$$R_b = \{v \in V(T) \setminus A \mid (v, b) \in E(T) \text{ and } |N^+(v) \cap B| \geq m_1(k)\},$$

$$S_b = \{v \in V(T) \setminus A \mid (v, b) \in E(T) \text{ and } |N^+(v) \cap B| \leq m_1(k) - 1\},$$

and $B_\emptyset = \{b \in B \mid S_b = \emptyset\}$. If $|B_\emptyset| \geq x(k)$, we claim that we can choose $X = B_\emptyset$.

To prove this claim, let $b \in B_\emptyset$ and assume that there is a path $P_i \in \mathcal{P}$ that accesses b from a vertex $v \in V(T) \setminus A$. With the choice of $m_1(k)$, we apply [Corollary 27](#) to conclude that v has at least $4k$ out-neighbors in $B \setminus X$ which are not used by paths of \mathcal{P} , and [Lemma 26](#) implies that at least $4k$ pairs $\{u, \mathbf{M}(u)\}$ with $u \in C$ are j -free for any $j \in [k]$. Thus we can choose an i -free vertex $w \in N^+(v) \cap B \setminus X$ and an i -free pair $\{u, \mathbf{M}(u)\}$ to reroute P_i through $v \rightarrow w \rightarrow u \rightarrow \mathbf{M}(u) \rightarrow b$. By [Corollary 27](#) at most $4k$ reroutings are needed since each P_i can enter X at most four times, and each rerouting can be done through distinct choices of w and $\{u, \mathbf{M}(u)\}$. Note that it is important to push this rerouting to vertices outside of X . If, for instance, $w \in X$, we would not be decreasing how many accesses from outside of the triple are done to X , essentially voiding our rerouting.

Assume now that $|B_\emptyset| \leq x(k) - 1$ and let $B_S = B \setminus B_\emptyset$. We build an auxiliary semicomplete digraph Γ with vertex set B_S as follows. For every ordered pair $\{b, b'\} \subseteq B_S$, we add (b, b') to $E(\Gamma)$ if, for all $v \in S_b$, it holds that $v \in S_{b'}$ or there is a $w \in S_{b'}$ such that $(v, w) \in E(T)$. Intuitively, the goal is to use arcs of Γ to identify when we can reroute a path entering b from outside of \mathcal{K} through b' . To see that Γ is indeed semicomplete, let $b, b' \in V(\Gamma)$. If there is no arc from b to b' , then some $v \in S_b \setminus S_{b'}$ is an in-neighbor in T of every vertex in $S_{b'} \setminus S_b$. By definition $(b', b) \in E(\Gamma)$ and therefore Γ is semicomplete. The definition of Γ immediately implies that it can be built in time $\mathcal{O}(f^2(k) \cdot n^2)$.

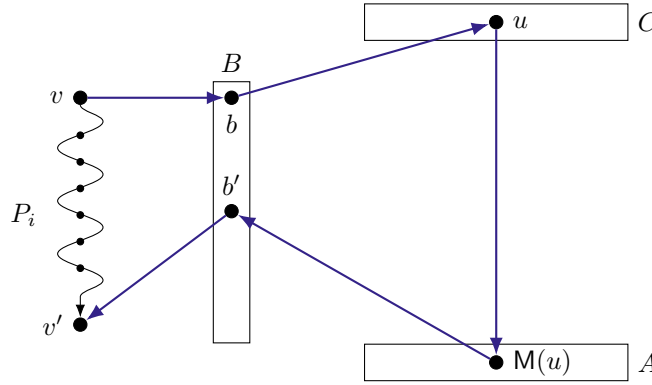
Now, the goal is to find a set of $x(k)$ vertices of $V(\Gamma)$ which have out-degree at least $d_1(k) \cdot m_1(k)$ in Γ . By discarding B_\emptyset from B , we conclude that $|V(\Gamma)| \geq |B| - x(k) + 1 = f(k) - x(k) + 1$. Assume that at most $x(k) - 1$ vertices of Γ have degree at least $d_1(k) \cdot m_1(k)$. Then Γ has at most $\alpha = (x(k) - 1)(|V(\Gamma)| - 1) + (|V(\Gamma)| - x(k) + 1)(d_1(k) \cdot m_1(k) - 1) \geq (x(k) - 1)(f(k) - x(k)) + (f(k) - 2x(k) + 2)(d_1(k) \cdot m_1(k) - 1)$ arcs. Thus by choosing $f(k)$ large enough to guarantee that $\binom{|V(\Gamma)|}{2} > \alpha$, we guarantee that at least $x(k)$ vertices of Γ have the desired out-degree. Let X contain all $b \in V(\Gamma)$ such that $\deg_\Gamma^+(v) \geq d_1(k) \cdot m_1(k)$.

We now prove that paths of \mathcal{P} accessing X from $V(T) \setminus A$ can be rerouted in the desired way. To achieve this, we introduce another shortcutting argument for \mathcal{P} .

▷ **Claim 28.** Let $W = \bigcup_{b \in B} S_b$ and M^+ be a matching from W to B . Then for all $i \in [k]$, the path P_i uses at most $7(4k+1) - 1$ vertices which are tails of arcs in M^+ .

Proof of the claim. By contradiction, assume that there is a $P_i \in \mathcal{P}$ using at least $7(4k+1)$ vertices which are tails of arcs in M^+ , and let W' denote the set of all such vertices. With $r = 4k+1$, we split W' arbitrarily into $4k$ sets W'_1, \dots, W'_{r-1} of size seven, plus one set W'_r with the remaining vertices of W' not included in any of the other sets; note that $|W'_r| \geq 7$.

By [Corollary 27](#), there is one set W'_j such that all heads of arcs of M^+ with tail in W'_j are i -free: at most $4k$ vertices of B are not i -free, so it follows that at most $4k$ different W'_i 's have at least one head in this set. Let v, v' be the first and last vertices of W'_j with respect to \prec_i and let b be the head of the arc $(v, b) \in M^+$. Since $v' \in S_b$ it holds that $|N_T^-(v') \cap B| \geq |B| - m_1(k) - 1 \geq 4k+1$. Applying [Lemma 26](#) and [Corollary 27](#) we can shortcut P_i through $v \rightarrow b \rightarrow u \rightarrow M(u) \rightarrow b' \rightarrow v'$, where b' is an i -free vertex of B and $\{u, M(u)\}$ is an i -free pair with $u \in C$. Recall that this is a contradiction since we are assuming that \mathcal{P} is of minimum total length, and so we have proved the claim. See [Figure 12](#) for an illustration of this shortcut. ◀



■ **Figure 12** In blue, a shortcut for a path P_i using at least $7(4k+1)$ vertices of $W = \bigcup_{b \in B} S_b$.

Assume that there is a $P_i \in \mathcal{P}$ such that P_i accesses some $b \in X$ through a vertex $v \in V(T) \setminus A$. If $v \in R_b$ then we proceed as if $S_b = \emptyset$: the out-degree of v in B is large enough to reroute through a vertex of B unoccupied by any path. Otherwise, $v \in S_b$. In this case, we use the bound on the degree of b in Γ and apply [Claim 28](#) to find a large matching in T from $V(T) \setminus A$ to B that can be used to reroute P_i .

We build in T a matching Y_0 from $W = \bigcup_{b \in B} S_b$ to B such that $Y_0 = \{(v, b), (v_1, b_1), \dots, (v_z, b_z)\}$ and:

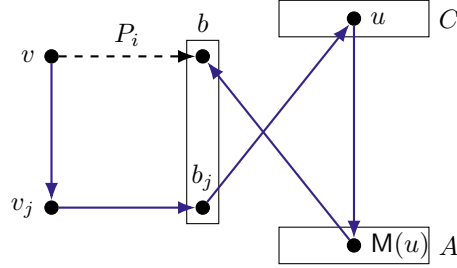
1. for $i \in [z]$ it holds that $(v, v_i) \in E(T)$, and
2. $z \geq d_1(k)$.

We say that (v, b) is the *distinguished* arc of Y_0 .

Let us show how to build Y_0 . We start with a collection \mathcal{W} of all sets $S_{b'}$ with $b' \in N_\Gamma^+(b)$ which are the candidates from which we can pick v_1 ; recall that $\deg_\Gamma^+(b) \geq d_1(k) \cdot m_1(k)$. First we trim from \mathcal{W} all such sets containing v , which are at most $m_1(k) - 1$ as $v \in S_b$. By definition, at most $m_1(k) - 1$ sets are discarded in this way. Choose b_1 such that $S_{b_1} \in \mathcal{W}$. The construction of Γ and the fact that $(b, b_1) \in E(\Gamma)$ imply that there is a $v_1 \in S_{b_1}$ such that

$(v, v_1) \in E(T)$. We add (v_1, b_1) to Y_0 and iterate. Assume that ℓ arcs $(v_1, b_1), \dots, (v_\ell, b_\ell)$ have been chosen this way, with $\ell \in [d_1(k) - 1]$. Thus, after discarding from \mathcal{W} every $S_{b'}$ containing any v_j with $j \in [\ell]$, we conclude that at least $m_1(k) \cdot (d_1(k) - \ell) + \ell$ candidates remain in \mathcal{W} for the choice of $v_{\ell+1}$. Since $\deg_\Gamma^+(b) \geq d_1(k) \cdot m_1(k)$, this procedure does not end before $d_1(k)$ arcs are chosen, and therefore we have constructed Y_0 with the desired properties. Notice that Y_0 can be built by simply observing the degrees of vertices in T , which is given in the input, and Γ , which can be built in polynomial time, and verifying which vertices are in each of the sets S_b , which can be done in time $\mathcal{O}(f^2(k) \cdot n^2)$.

We now show that for some $j \in [d_1(k)]$ the arc $(v_j, b_j) \in Y_0$ has both its endpoints i -free. By Claim 28, at most $7k(4k+1)$ vertices which are heads of arcs in Y_0 are used by paths of \mathcal{P} . By Corollary 27, at most $4k$ arcs of Y_0 have their endpoints in B being used by a path of \mathcal{P} . Thus, the choice of $d_1(k)$ guarantees that there are at least $4k$ arcs of the form $(v_j, b_j) \in Y_0$ with $j \in [d_1(k)]$ with both its endpoints being i -free; the $+8k$ term in the definition of $d_1(k)$ is present precisely so we can claim that these $4k$ arcs exist throughout the first rerouting phase. This suffices to reroute the path P_i through $v \rightarrow v_j \rightarrow b_j \rightarrow u \rightarrow M(u) \rightarrow b$, where $\{u, M(u)\}$ is an i -free pair with $u \in C$. See Figure 13 for an illustration of this rerouting.



■ **Figure 13** In blue, the rerouting for the path P_i (dashed) accessing b from $V(T) \setminus A$.

Since $|X| = x(k)$, for each other $b' \in X$ we can do a similar rerouting as described above while the choice of $d_1(k)$ allows us to, in addition, avoid rerouting any paths through distinguished arcs (v'', b'') with $b'' \in X$. Again by Corollary 27, at most $4k$ reroutings are done in this case as well, and therefore the choice of $d_1(k)$ guarantees that there is room for all reroutings.

We denote by \mathcal{Q} our new solution built from \mathcal{P} after this first round of reroutings. Essentially, after the first round of reroutings we have that \mathcal{Q} uses at most $4k$ new vertices of B and $4k$ new pairs $\{u, M(u)\}$, with $u \in C$, when compared with \mathcal{P} , and all reroutings are done through vertices outside of X .

Second round of reroutings. The goal now is to find a *special* vertex $b \in X$ such that if there is a solution for \mathcal{I} , then there is a solution $\mathcal{Q}' = \{Q'_1, \dots, Q'_k\}$ such that:

- (a) any path $Q'_i \in \mathcal{Q}'$ leaving b does so through a vertex $u \in C$, and
- (b) at most c new vertices of B and at most c new pairs $\{u, M(u)\}$ are used by \mathcal{Q}' in comparison with \mathcal{Q} .

The remaining part of the proof is mostly similar and easier than the previous one. The main difference is that at this point Property 23 no longer holds with respect to \mathcal{Q} . Hence, in order to obtain a similar result as in Claim 28, we use the third defining property of X : it uses at most one new vertex of $V(T) \setminus V(\mathcal{K})$ in comparison with \mathcal{P} for each of the k paths. For every $b \in B$, let:

$$R'_b = \{v \in V(T) \setminus A \mid (v, b) \in E(T) \text{ and } |N_T^-(v) \cap B| \geq m_2(k)\},$$

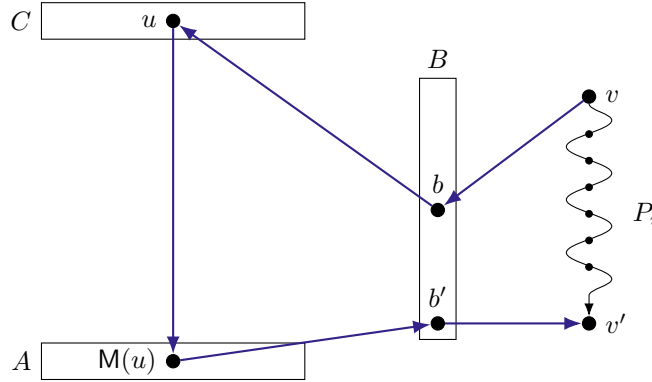
$$S'_b = \{v \in V(T) \setminus A \mid (v, b) \in E(T) \text{ and } |N_T^-(v) \cap B| \leq m_2(k) - 1\}.$$

▷ **Claim 29.** Let $W = \bigcup_{b \in B} S'_b$ and M^- be a matching from B to W . Then for all $i \in [k]$, the path Q_i uses at most $8(4k+1) - 1$ vertices which are heads of arcs in M^- .

Proof of the claim. By contradiction, assume that there is a $Q_i \in \mathcal{Q}$ using at least $8(4k+1)$ vertices which are heads of arcs in M^- , and let W' denote the set of all such vertices. With $r = 4k+1$, we split W' arbitrarily into $4k$ sets W'_1, \dots, W'_{r-1} of size eight, plus one set W'_r with the remaining vertices of W' not included in any of the other sets; note that $|W'_r| \geq 8$.

By [Corollary 27](#), there is one set W'_j such that all tails of arcs of M^- with heads in W'_j are not used by any path in \mathcal{Q} . We consider two cases.

If $Q_i = P_i$, then we can shortcut P_i similarly to what is done in the proof of [Claim 28](#), contradicting [Property 23](#). Denote by v, v' the first and last vertices of W'_j with respect to \prec_i , and let b' be the tail of the arc $(b', v') \in M^-$. Since v is in some S'_b , it holds that $|N_T^+(v) \cap B| \geq |B| - m_2(k) + 1 \geq 4k+1$. Thus, by [Lemma 26](#) and [Corollary 27](#) we can shortcut P_i through $v \rightarrow b \rightarrow u \rightarrow M(u) \rightarrow b' \rightarrow v'$, where b is an i -free vertex in B and $\{u, M(u)\}$ is an i -free pair with $u \in C$, and the claim follows. See [Figure 14](#) for an illustration of this shortcut.



■ **Figure 14** In blue, a shortcut for a path P_i associated with a path Q_i which uses at least $8(4k+1)$ vertices of $W = \bigcup_{b \in B} S'_b$.

Assume now that Q_i was built by rerouting the path P_i . By property (iii) of X , at least seven vertices of W'_j are used by P_i . In this case we can shortcut P_i similarly to the first case, and the claim follows. ◀

Assume that there is a $b \in B$ such that $S'_b = \emptyset$ and that there is a path Q_i leaving b through a vertex $v \in V(T) \setminus C$. Thus $v \in R'_b$ and therefore $|N_T^-(v) \cap B| \geq m_2(k)$. By [Corollary 27](#) this implies that v has in-neighbors $b_1, \dots, b_c \in B$ such that no b_j with $j \in [c]$ is used by a path of \mathcal{P} and was not used by any of the reroutings from the first round. Thus, due to [Lemma 26](#), we can reroute Q_i through $b \rightarrow u \rightarrow M(u) \rightarrow b_1 \rightarrow v$, where $\{u, M(u)\}$ is an i -free pair that was not used by any of the reroutings from the first round. Since at most c paths of Q_i can use b , there is room in b_1, \dots, b_c to reroute all paths of \mathcal{Q} using b . In this case, b is a good choice for the special vertex and the second round of reroutings is finished.

Assume now that every $b \in B$ has $S'_b \neq \emptyset$. We construct an auxiliary semicomplete digraph Γ' with vertex set X by adding to it the arc (b', b) if, for all $v \in S'_b$, either $v \in S'_{b'}$ or there is an arc from a $v' \in S'_{b'}$ to v . The proof that Γ' is indeed semicomplete is analogous to the proof for Γ , and it can also be constructed in time $\mathcal{O}(f^2(k) \cdot n^2)$.

Since $|X| = 2(m_2(k) \cdot d_2(k))$, it holds that there is a vertex $b \in V(\Gamma')$ such that $\deg_{\Gamma'}^-(b) \geq m_2(k) \cdot d_2(k)$. We show that b is a good choice for the special vertex. To do so, assume that there is a path $Q_i \in \mathcal{Q}$ that leaves b through a vertex $v \in V(T) \setminus C$. If $v \in R'_b$ we proceed as in the previous case of this round. Assume now that $v \in S'_b$.

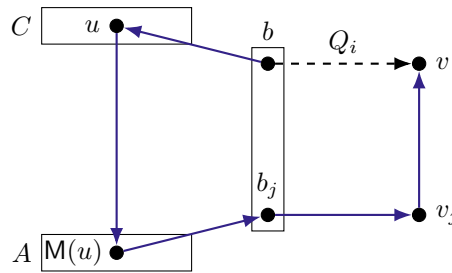
We build in T a matching Y' from B to $W' = \bigcup_{b \in B} S'_b$ such that $Y' = \{(v, b), (v_1, b_1), \dots, (v_z, b_z)\}$ and:

1. for $i \in [z]$ it holds that $(v_i, v) \in E(T)$, and
2. $z \geq d_2(k)$.

We start with a collection \mathcal{W}' of all sets $S'_{b'}$ with $b' \in N_{\Gamma'}^-(b)$ which are the candidates from which we can pick v_1 . We trim from \mathcal{W}' all such sets containing v . By definition, at most $m_2(k) - 1$ sets are discarded in this way. Choose any b_1 such $S'_{b_1} \in \mathcal{W}'$. The construction of Γ' and the fact that $(b_1, b) \in E(\Gamma')$ implies that there is a $v_1 \in S'_{b_1}$ such that $(v_1, v) \in E(T)$. We add (b_1, v_1) to Y' .

For the iterative argument, assume that ℓ arcs $(b_1, v_1), \dots, (b_\ell, v_\ell)$ have been chosen this way, with $\ell \in [d_2(k) - 1]$. Thus, after discarding from \mathcal{W}' every $S'_{b'}$ containing any v_j with $j \in [\ell]$, we conclude that at least $m_2(k) \cdot (d_2(k) - \ell) + \ell$ candidates for the choice of $v_{\ell+1}$ remain in \mathcal{W}' . Since $\deg_{\Gamma'}^-(b) \geq d_2(k) \cdot m_2(k)$, this procedure does not end before $d_2(k)$ arcs are chosen this way. Therefore, we terminate with the desired set Y' which, similar to the similar procedure in the first round of reroutings, can be built in $\mathcal{O}(f^2(k) \cdot n^2)$ time.

We show that the choice of $d_2(k)$ implies that at least one $(b_j, v_j) \in Y'$ has both its endpoints available to reroute Q_i . By Claim 29 at most $8k(4k + 1)$ vertices which are tails of arcs in Y' are used by paths of \mathcal{Q} . We eliminate from Y' all arcs associated with those vertices. By Corollary 27, at most $4k$ arcs of Y' have their endpoints in B being used by paths of \mathcal{P} . By property (iii) of X , the endpoints in B of at most $4k$ arcs of Y' were used by reroutings from the first round. We eliminate from Y' all arcs from those two sets. In the end, at least c arcs remain in Y' . Let (b_j, v_j) be one of those remaining arcs. Then we can reroute Q_i through $b \rightarrow u \rightarrow M(u) \rightarrow b_j \rightarrow v_j \rightarrow v$, as shown in Figure 15. Since at least c arcs remained in Y' and at most c paths of \mathcal{Q} use b , we can reroute any path leaving b to a vertex in $V(T) \setminus A$ similarly. After the second round of reroutings, we choose b as the special vertex.



■ **Figure 15** In blue, the rerouting for the path Q_i (dashed) leaving b to $V(T) \setminus A$.

Finding the irrelevant vertex. It is easy to see that b is an irrelevant vertex. By property (b) there are c vertices in B which are not used by any path of \mathcal{Q}' . Since any path Q'_i using b reaches this vertex from a $v \in A$ and leaves it through a $u \in C$, it suffices to reroute Q'_i using $v \rightarrow b' \rightarrow u$, where b' is any vertex of B not used by paths of \mathcal{Q}' . Since there are at least c such vertices, all paths of \mathcal{Q}' can be rerouted away from b and the result follows. ◀

Note that Lemmas 24–26 and Corollary 27 only deal with arcs and reroutings *inside* of the $f(k)$ -triple \mathcal{K} . Thus, in order to adapt the proof to h -semicomplete digraphs, it suffices to add a factor of h to the steps above where vertices of large in- or out-neighborhood in B or in the heads or tails of the matchings of the form Y_0 (after Claim 28) and Y' (after Claim 29), for example. This implies that at each of those steps, the number of non-neighbors of the observed vertex is taken into account, and thus the same ideas work for $h \geq 1$.

6 Future research

This work touched on several common strategies used to cope with the hardness of the DIRECTED DISJOINT PATHS problem. We investigated its (parameterized) complexity on tournaments and some of its superclasses, both with and without the typical relaxation of allowing congestion in the vertices. One of our key contributions is a fix (Theorem 1) for a gap in the literature originating in a flaw in the NP-hardness proof for k -DDP on tournaments given in [2]; we also adapt a win/win approach based on directed pathwidth first used in an FPT algorithm for DIRECTED EDGE-DISJOINT PATHS parameterized by the number of paths on tournaments [23] to (k, c) -DDP when $2c > k$. We note, however, that this latter problem is *not* known to be NP-complete; while we are able to show that (k, k^ε) -DDP is hard for every $\varepsilon \in [0, 1)$, we are unable to extend this to $c = \varepsilon k$, or even $c = k/2 + 1$. Alongside the already challenging problems listed in Section 1, we consider this the main open question related to our work. We do not provide all the details for the proof of Theorem 20 in its full generality, only for the case of semicomplete graphs, i.e., $h = 0$. Extending our arguments to $h > 0$ is essentially going deeper into technicalities: we must increase the size of the triple taking h into account and, when discussing the exterior neighborhood of the triple, increase the thresholds to classify the in- and out-neighbors of B as having “too many” neighbors in B or not. This extension, however, is not enough to give an FPT algorithm for (k, c) -DDP on h -semicomplete graphs when $2c > k$. In particular, two challenges remain: (i) computing a triple for elements of this class in FPT-time, as the only known algorithm being an XP one introduced by Kitsunai, Kobayashi, and Tamaki [36]; and (ii) devising an FPT algorithm for the joint parameterization by k , h , and the directed pathwidth, which would also improve upon the XP algorithm shown in [36]. We recall that, among the classes studied in our work, an FPT algorithm for h -semicomplete digraphs is the best we can hope for, as we have shown that k -DDP and (k, c) -DDP are W[1]-hard when parameterized by k and the number of covering tournaments on digraphs of directed pathwidth two (Theorem 4 and Theorem 19).

References

- 1 Saeed A. Amiri, Stephan Kreutzer, Dániel Marx, and Roman Rabinovich. Routing with congestion in acyclic digraphs. *Information Processing Letters*, 151:105836, 2019. doi: doi.org/10.1016/j.ipl.2019.105836.
- 2 Jørgen Bang-Jensen and Carsten Thomassen. A polynomial algorithm for the 2-path problem for semicomplete digraphs. *SIAM Journal on Discrete Mathematics*, 5(3):366–376, 1992. arXiv: <https://doi.org/10.1137/0405027>, doi: [10.1137/0405027](https://doi.org/10.1137/0405027).
- 3 Jørgen Bang-Jensen and Carsten Thomassen. Personal communication, 2024.
- 4 Jørgen Bang-Jensen and Gregory Gutin. *Classes of Directed Graphs*. Springer Monographs in Mathematics, 2018. doi: [10.1007/978-3-319-71840-8](https://doi.org/10.1007/978-3-319-71840-8).
- 5 Florian Barbero, Christophe Paul, and Michał Pilipczuk. Strong immersion is a well-quasi-ordering for semicomplete digraphs. *Journal of Graph Theory*, 90(4):484–496, 2019. doi: [10.1002/JGT.22408](https://doi.org/10.1002/JGT.22408).

- 6 Adrian Bondy and U.S.R. Murty. *Graph Theory*. Springer-Verlag London, 2008.
- 7 Ulrik Brandes, Wolfram Schlickerrieder, Gabriele Neyer, Dorothea Wagner, and Karsten Weihe. A software package of algorithms and heuristics for disjoint paths in planar networks. *Discrete Applied Mathematics*, 92(2-3):91–110, 1999. doi:[10.1016/S0166-218X\(99\)00048-7](https://doi.org/10.1016/S0166-218X(99)00048-7).
- 8 Rowland Leonard Brooks. On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37(2):194–197, 1941. doi:[10.1017/S030500410002168X](https://doi.org/10.1017/S030500410002168X).
- 9 Victor Campos, Raul Lopes, Ana Karolinna Maia, and Ignasi Sau. Adapting the Directed Grid Theorem into an FPT Algorithm. *SIAM Journal on Discrete Mathematics*, 36(3):1887–1917, 2022. doi:[10.1137/21M1452664](https://doi.org/10.1137/21M1452664).
- 10 Victor A. Campos, Jonas Costa, Raul Lopes, and Ignasi Sau. New Menger-Like Dualities in Digraphs and Applications to Half-Integral Linkages. In *Proc. of the 31st Annual European Symposium on Algorithms (ESA)*, volume 274 of *LIPICs*, pages 30:1–30:18, 2023. doi:[10.4230/LIPICs.ESA.2023.30](https://doi.org/10.4230/LIPICs.ESA.2023.30).
- 11 Dario Giuliano Cavallaro, Ken-ichi Kawarabayashi, and Stephan Kreutzer. Edge-Disjoint Paths in Eulerian Digraphs. In *Proc. of the 56th Annual ACM Symposium on Theory of Computing (STOC)*, pages 704–715, 2024. doi:[10.1145/3618260.3649758](https://doi.org/10.1145/3618260.3649758).
- 12 Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. An $o(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, 2(7):137–146, 2006. doi:[10.4086/TOC.2006.V002A007](https://doi.org/10.4086/TOC.2006.V002A007).
- 13 Maria Chudnovsky, Alexandra O. Fradkin, and Paul D. Seymour. Tournament immersion and cutwidth. *Journal of Combinatorial Theory, Series B*, 102(1):93–101, 2012. doi:[10.1016/J.JCTB.2011.05.001](https://doi.org/10.1016/J.JCTB.2011.05.001).
- 14 Maria Chudnovsky, Alex Scott, and Paul Seymour. Disjoint paths in unions of tournaments. *Journal of Combinatorial Theory, Series B*, 135:238–255, 2019. URL: <https://www.sciencedirect.com/science/article/pii/S0095895618300789>, doi:[10.1016/j.jctb.2018.08.007](https://doi.org/10.1016/j.jctb.2018.08.007).
- 15 Maria Chudnovsky and Paul D. Seymour. A well-quasi-order for tournaments. *Journal of Combinatorial Theory, Series B*, 101(1):47–53, 2011. doi:[10.1016/J.JCTB.2010.10.003](https://doi.org/10.1016/J.JCTB.2010.10.003).
- 16 Maria Chudnovsky, Paul D. Seymour, and Alex Scott. Disjoint paths in tournaments. *Advances in Mathematics*, 270:582–597, 2015. doi:[10.1016/j.aim.2014.11.011](https://doi.org/10.1016/j.aim.2014.11.011).
- 17 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:[10.1007/978-3-319-21275-3](https://doi.org/10.1007/978-3-319-21275-3).
- 18 Andreas Darmann and Janosch Döcker. On simplified np-complete variants of monotone 3-sat. *Discrete Applied Mathematics*, 292:45–58, 2021. URL: <https://www.sciencedirect.com/science/article/pii/S0166218X2030528X>, doi:[10.1016/j.dam.2020.12.010](https://doi.org/10.1016/j.dam.2020.12.010).
- 19 Rod G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- 20 Katherine Edwards, Irene Muzi, and Paul Wollan. Half-Integral Linkages in Highly Connected Directed Graphs. In *Proc. of the 25th Annual European Symposium on Algorithms (ESA)*, volume 87 of *LIPICs*, pages 36:1–36:12, 2017. Full version available in <https://arxiv.org/abs/1611.01004>. doi:[10.4230/LIPICs.ESA.2017.36](https://doi.org/10.4230/LIPICs.ESA.2017.36).
- 21 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006. doi:[10.1007/3-540-29953-X](https://doi.org/10.1007/3-540-29953-X).
- 22 Fedor V. Fomin, Daniel Lokshantov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019. doi:[10.1017/9781107415157](https://doi.org/10.1017/9781107415157).
- 23 Fedor V. Fomin and Michal Pilipczuk. On width measures and topological problems on semi-complete digraphs. *Journal of Combinatorial Theory, Series B*, 138:78–165, 2019. doi:[10.1016/J.JCTB.2019.01.006](https://doi.org/10.1016/J.JCTB.2019.01.006).

- 24 Fedor V. Fomin and Michał Pilipczuk. On width measures and topological problems on semi-complete digraphs. *Journal of Combinatorial Theory, Series B*, 138:78–165, 2019. doi:[10.1016/j.jctb.2019.01.006](https://doi.org/10.1016/j.jctb.2019.01.006).
- 25 Steven Fortune, John E. Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10:111–121, 1980. doi:[10.1016/0304-3975\(80\)90009-2](https://doi.org/10.1016/0304-3975(80)90009-2).
- 26 Alexandra O. Fradkin and Paul D. Seymour. Tournament pathwidth and topological containment. *Journal of Combinatorial Theory, Series B*, 103(3):374–384, 2013. doi:[10.1016/J.JCTB.2013.03.001](https://doi.org/10.1016/J.JCTB.2013.03.001).
- 27 Alexandra O. Fradkin and Paul D. Seymour. Edge-disjoint paths in digraphs with bounded independence number. *Journal of Combinatorial Theory, Series B*, 110:19–46, 2015. doi:[10.1016/J.JCTB.2014.07.002](https://doi.org/10.1016/J.JCTB.2014.07.002).
- 28 Robert Ganian, Petr Hliněný, Joachim Kneis, Alexander Langer, Jan Obdržálek, and Peter Rossmanith. Digraph width measures in parameterized algorithmics. *Discrete Applied Mathematics*, 168:88–107, 2014. doi:[10.1016/j.dam.2013.10.038](https://doi.org/10.1016/j.dam.2013.10.038).
- 29 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. URL: <https://dl.acm.org/doi/10.5555/574848>.
- 30 Archontia C. Giannopoulou, Ken-ichi Kawarabayashi, Stephan Kreutzer, and O-joung Kwon. The canonical directed tree decomposition and its applications to the directed disjoint paths problem, 2020. This article is the full version of reference [31]. arXiv:2009.13184.
- 31 Archontia C. Giannopoulou, Ken-ichi Kawarabayashi, Stephan Kreutzer, and O-joung Kwon. Directed tangle tree-decompositions and applications. In *Proc. of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 377–405, 2022. The full version of this article is reference [30]. doi:[10.1137/1.9781611977073.19](https://doi.org/10.1137/1.9781611977073.19).
- 32 Meike Hatzel, Stephan Kreutzer, Marcelo Garlet Milani, and Irene Muzi. Cycles of Well-Linked Sets and an Elementary Bound for the Directed Grid Theorem. In *Proc. of the 65th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–20, 2024. doi:[10.1109/FOCS61266.2024.00011](https://doi.org/10.1109/FOCS61266.2024.00011).
- 33 Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Stephan Kreutzer. An excluded half-integral grid theorem for digraphs and the directed disjoint paths problem. In *Proc. of the 46th Annual ACM on Symposium on Theory of Computing (STOC)*, pages 70–78, 2014. doi:[10.1145/2591796.2591876](https://doi.org/10.1145/2591796.2591876).
- 34 Ken-ichi Kawarabayashi and Stephan Kreutzer. The Directed Grid Theorem. In *Proc. of the 47th Annual ACM on Symposium on Theory of Computing (STOC)*, pages 655–664. ACM, 2015. doi:[10.1145/2746539.2746586](https://doi.org/10.1145/2746539.2746586).
- 35 Ilhee Kim and Paul D. Seymour. Tournament minors. *Journal of Combinatorial Theory, Series B*, 112:138–153, 2015. doi:[10.1016/J.JCTB.2014.12.005](https://doi.org/10.1016/J.JCTB.2014.12.005).
- 36 Kenta Kitsunai, Yasuaki Kobayashi, and Hisao Tamaki. On the pathwidth of almost semi-complete digraphs. In Nikhil Bansal and Irene Finocchi, editors, *Proc. of the 23rd Annual European Symposium on Algorithms (ESA)*, volume 9294 of *LNCS*, pages 816–827, 2015. doi:[10.1007/978-3-662-48350-3_68](https://doi.org/10.1007/978-3-662-48350-3_68).
- 37 Raul Lopes and Ignasi Sau. A relaxation of the Directed Disjoint Paths problem: A global congestion metric helps. *Theoretical Computer Science*, 898:75–91, 2022. doi:[10.1016/j.tcs.2021.10.023](https://doi.org/10.1016/j.tcs.2021.10.023).
- 38 Lúcia Martins, Teresa Gomes, and David Tipper. Efficient heuristics for determining node-disjoint path pairs visiting specified nodes. *Networks*, 70(4):292–307, 2017. doi:[10.1002/NET.21778](https://doi.org/10.1002/NET.21778).
- 39 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. doi:[10.1093/acprof:oso/9780198566076.001.0001](https://doi.org/10.1093/acprof:oso/9780198566076.001.0001).
- 40 Neil Robertson and Paul D. Seymour. Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. doi:[10.1006/JCTB.1995.1006](https://doi.org/10.1006/JCTB.1995.1006).

- 41 Aleksandrs Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. *SIAM Journal on Discrete Mathematics*, 24(1):146–157, 2010. doi:[10.1137/070697781](https://doi.org/10.1137/070697781).
- 42 Carsten Thomassen. Highly connected non-2-linked digraphs. *Combinatorica*, 11(4):393–395, 1991. doi:[10.1007/BF01275674](https://doi.org/10.1007/BF01275674).
- 43 Michal Włodarczyk. Constant Approximating Disjoint Paths on Acyclic Digraphs Is W[1]-Hard. In *Proc. of the 35th International Symposium on Algorithms and Computation (ISAAC)*, volume 322 of *LIPICs*, pages 57:1–57:16, 2024. doi:[10.4230/LIPICS.ISAAC.2024.57](https://doi.org/10.4230/LIPICS.ISAAC.2024.57).

A Flaw in the NP-completeness proof of Bang-Jensen and Thomassen

In this appendix we sketch the flaw in the proof of Bang-Jensen and Thomassen claiming that DIRECTED DISJOINT PATHS is NP-complete on tournaments when k is part of the input.

Their approach is to first prove the NP-completeness of a related problem on tournaments, and then reduce from this problem to DIRECTED DISJOINT PATHS. Namely, it is proved in [2, Theorem 6.1] that the following problem Q is NP-complete on tournaments: given a tournament T and a set of arcs $A \subseteq E(T)$, decide whether T contains a directed cycle visiting all the arcs in A (in any order). The proof of [2, Theorem 6.1] is correct, and consists in a simple reduction from HAMILTONIAN CYCLE on general digraphs to Q on tournaments.

The problem comes later: right after the proof of [2, Theorem 6.1], it is claimed that “This proves that, if k is not fixed, then the k -DDP problem is NP-complete on tournaments”. This statement is not clear at all, the main issue being that in problem Q the set of arcs A to be visited by the cycle is *unordered*. Indeed, in order to construct an instance of k -DDP starting from an instance (T, A) of problem Q , the natural strategy would be to consider an arbitrary ordering $\sigma = (e_0, \dots, e_{k-1})$ of the arcs in A , and construct a set K of k requests in T as follows. For every $i \in \{0, \dots, k-1\}$, add to K a request from the head of e_i to the tail of e_{i+1} , where indices are taken modulo k . One may hope that the desired cycle C in T visiting all the arcs in A would translate to the existence of k pairwise vertex-disjoint paths in T satisfying all the requests in K . But this is not true, as C may exist in T , but may yield a visiting ordering of the arcs in A that is different from the ordering σ that we have fixed arbitrarily. One could try to fix this issue by guessing all the possible orderings σ of the arcs in A , but this would result in $k!$ choices, which is not allowed in a polynomial-time reduction since k is considered as part of the input.

We have shared the above issue with the authors of [2] and they have confirmed to us [3] that it does not seem to be easily fixable.