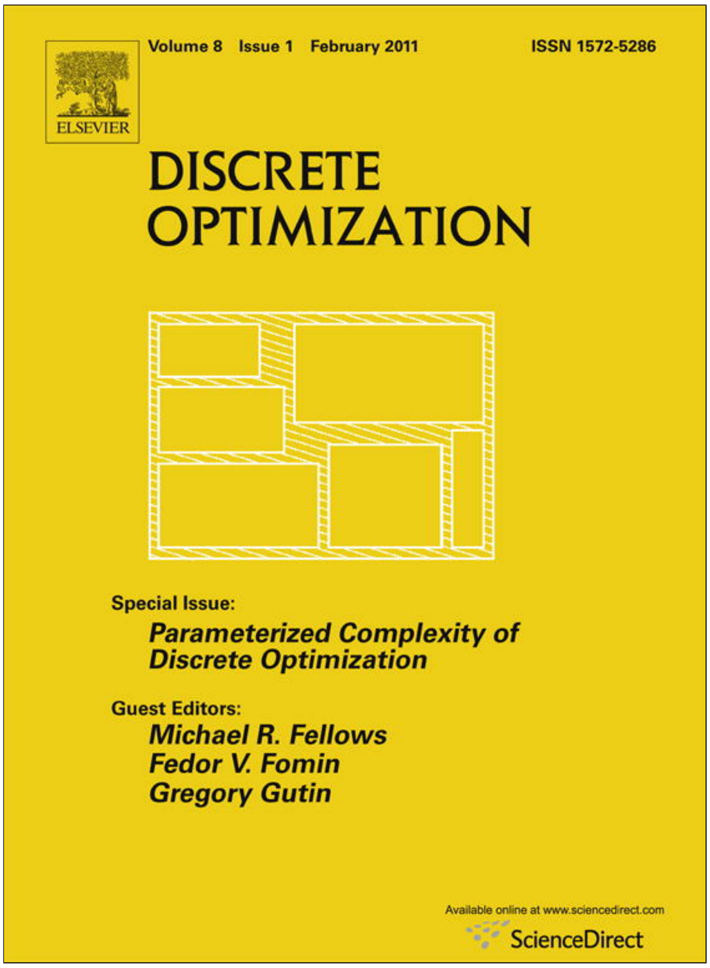

Volume 8 Issue 1 February 2011
ISSN 1572-5286
ELSEVIER
DISCRETE OPTIMIZATION
Special Issue:
Parameterized Complexity of Discrete Optimization
Guest Editors:
Michael R. Fellows
Fedor V. Fomin
Gregory Gutin
Available online at www.sciencedirect.com
ScienceDirect

Discrete Optimization 8 (2011) 72–86

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

Paths of bounded length and their cuts: Parameterized complexity and algorithms[☆]

Petr A. Golovach[a], Dimitrios M. Thilikos[b,*]

[a] School of Engineering and Computing Sciences, Durham University, South Road, Durham DH1 3LE, UK

[b] Department of Mathematics, National and Kapodistrian University of Athens, Panepistimioupolis, GR15784 Athens, Greece

ARTICLE INFO

Article history:
Received 13 November 2009
Received in revised form 21 August 2010
Accepted 27 September 2010
Available online 23 October 2010

Keywords:
Bounded length disjoint paths
Bounded length cuts
Parameterized complexity
Parameterized algorithms

ABSTRACT

We study the parameterized complexity of two families of problems: the *bounded length disjoint paths* problem and the *bounded length cut* problem. From Menger's theorem both problems are equivalent (and computationally easy) in the unbounded case for single source, single target paths. However, in the bounded case, they are combinatorially distinct and are both NP-hard, even to approximate. Our results indicate that a more refined landscape appears when we study these problems with respect to their parameterized complexity. For this, we consider several parameterizations (with respect to the maximum length l of paths, the number k of paths or the size of a cut, and the treewidth of the input graph) of all variants of both problems (edge/vertex-disjoint paths or cuts, directed/undirected). We provide FPT-algorithms (for all variants) when parameterized by both k and l and hardness results when the parameter is only one of k and l. Our results indicate that the bounded length disjoint-path variants are structurally harder than their bounded length cut counterparts. Also, it appears that the edge variants are harder than their vertex-disjoint counterparts when parameterized by the treewidth of the input graph.

1. Introduction and preliminaries

We consider finite (directed and undirected) graphs without loops or multiple edges. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$. We denote undirected edges by $\{u, v\}$, and directed edges by (u, v). Given a graph G and a set $F \subseteq E(G)$ (resp. $X \subseteq V(G)$), we denote by $G \setminus F$ (resp. $G \setminus X$) the graph obtained by G if we remove from it all edges in F (resp. vertices in X). The path in a graph G induced by vertices $u_1, \dots, u_k$ (in the given order) is denoted by $\langle u_1, \dots, u_k \rangle$. Concatenation of the paths P_1 and P_2 such the last vertex of P_1 coincides with the first vertex of P_2 is denoted by $P_1 \oplus P_2$. The distance between two vertices $u, v \in V(G)$ (i.e., the length of the shortest (u, v)-path in the graph) is denoted by $\mathrm{dist}_G(u, v)$.

One of the most celebrated problems in discrete algorithms and combinatorial optimization is the disjoint paths problem. Its algorithmic study dates back to Menger's theorem [1] (see also [2]), was extended by the work of Ford and Fulkerson [3] on network flows, and now constitutes (along with its variants) a central algorithmic problem in algorithm design.

According to Menger's theorem, given a graph G and two terminals $s, t \in V(G)$, the maximum number of vertex-disjoint (s, t)-paths in G is *equal* to the minimum cardinality of a set of vertices in $V(G) \setminus \{s, t\}$ meeting all (s, t)-paths of G. Interestingly, it appears that such a min–max equality does not hold if we restrict paths to be of bounded length. This was observed for the first time by Adámek and Koubek in [4]. Lovász et al. proved in [5] that a similar min–max relation

☆ Preliminary extended abstracts of this paper appeared in the proceedings of IWPEC'09 (Golovach and Thilikos (2009) [36]).

* Corresponding author.
E-mail address: sedthilk@math.uoa.gr (D.M. Thilikos).

1572-5286/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2010.09.009

Table 1
Bounded length variants of the disjoint paths problem and the cut problem.

	Multi-terminal disjoint Paths		(s,t)-disjoint Paths		(s,t)-Cut	
	Directed	Undirected	Directed	Undirected	Directed	Undirected
Edge	BEDMP	BEUMP	BEDP	BEUP	BEDC	BEUC
Vertex	BVDMP	BVUMP	BVDP	BVUP	BVDC	BVUC

holds only for path lengths equal to 2, 3, or 4. Analogous results were provided for the case where the paths are edge-disjoint in [6,7].

We present below the main decision versions of the problems generated by the bounded length restriction. We need some definitions. Let G be a graph, $s, t \in V(G)$, and let l be a positive integer. We call a set $F \subseteq E(G)$ (resp. $X \subseteq V(G) \setminus \{s, t\}$) a *$(s, t)$-edge* (resp. *vertex*) *l-bounded cut* if $G \setminus F$ (resp. $G \setminus X$) contains no (s, t)-path of length at most l.

Bounded Edge Directed (s, t)-disjoint Paths (BEDP)

Input: A directed graph G, two positive integers k, l and two distinct vertices s, t of G.
Question: Are there k edge-disjoint (s, t)-paths each of length at most l in G?

Bounded Edge Directed (s, t)-Cut (BEDC)

Input: A directed graph G, two positive integers k, l and two distinct vertices s, t of G.
Question: Is there an (s, t)-edge l-bounded cut $F \subseteq E(G)$ of size at most k?

Also, the first of the above problems has been extended to its multi-terminal version as follows.

Bounded Edge Directed Multi-terminal disjoint Paths (BEDMP)

Input: A directed graph $G = (V, E)$, two positive integers k, l, and two sequences $S = (s_1, \dots, s_k)$ (sources), $T = (t_1, \dots, t_k)$ (targets) of vertices in G.
Question: Are there k edge-disjoint (s_i, t_i)-paths of length at most l in G for $i = 1, \dots, k$?

Similar to the above, we can define numerous variants depending on whether the graph is directed or undirected, and whether the paths are edge-disjoint or internally vertex-disjoint. All variants and the corresponding notations are depicted in Table 1.

For all multi-terminal disjoint path problems we can assume that all terminals are pair-wise distinct, since otherwise we can apply the following rule:

Rule (1): for every vertex v that corresponds to r terminals we first subdivide all its incident edges and then replace v by r vertices (each with one of the terminals corresponding to v) that have the same neighborhood as v (in the directed case, replacement edges maintain their original directions). The new graph contains k edge(vertex)-disjoint paths of length at most $l + 2$ if and only if the original one contains k edge(vertex)-disjoint paths of length at most l.

The first algorithmic results for the above problems were presented by Itai et al. in [8] where they proved that BVUP and BEUP are polynomially solvable for path lengths 2 or 3 (for BVUP, it was proved also for paths of lengths at most 4), while they become NP-complete for length values bigger than 4. In the same paper they proved that if, instead of fixing the length l, we fix the number k of paths, the problem is still NP-complete even for 2 paths. For cut problems, Baier et al. [9,10] proved that BVUC, BVDC are NP-complete for length values at least 5, and BEUC, BEDC are NP-complete for path length bounds bigger than 3 (in fact they proved that it is NP-hard to approximate the size of the minimum cut within a factor of at least 1.1377). The approximability of these problems was studied in [9–12]. Particularly, Bley [11] proved that computing the maximum number of vertex-disjoint s, t-paths in an undirected graph is *APX*-complete for any $l \geq 5$. Results on the fractional versions of these problems (in terms of multicommodity flow problems) were given in [13–15]. Finally, for some applications of the above problems, see [16–18].

Some results for the multi-terminal variants of the bounded-length disjoint paths problem were given in [12]. We just stress that, when there is no restriction on the length of the paths, BVUMP is NP-complete in general [19] and polynomially solvable in cubic time for any fixed k [20], while BVDMP is NP-complete even when $k = 2$ [21].

In this paper, we provide a detailed study of the parameterized complexity of all the bounded length variants of the problems in Table 1. In a parameterized problem we distinguish some part of the input to be its parameter. Typically, a parameter is an integer, k, related to the problem input and the question is whether the problem can be solved by an algorithm (called an FPT-*algorithm*) of time complexity $f(k) \cdot n^{O(1)}$ where n is the size of the input and f is a (super-polynomial) function depending only on the parameter (instead of worst time complexities such as $O(n^{f(k)})$ or $O(k^{f(n)})$). When a parameterized problem admits an FPT-algorithm, then it belongs in the parameterized complexity class FPT. Not all parameterized problems belong in FPT. There are several parameterized complexity classes, such as W[1], W[2], para-NP, and analogous notions of hardness with respect to parameter-preserving reductions, able to prove that membership in FPT is rather non-possible (for more details, see the monographs [22–24]). Briefly, if a parameterized problem is W[1]-hard, this means that a complexity of type $O(n^{f(k)})$ is the best we may expect unless FPT = W[1], while if a parameterized problem is para-NP-hard, then we cannot even hope for something better than a $k^{f(n)}$-algorithm unless P = NP.

Table 2
Reductions between problems.

BEUMP	$\leq^{(2)}$	BVUMP	$\leq^{(3)}$	BVDMP	$\leq^{(4)}$	BEDMP
$\vee\vert^{(1)}$		$\vee\vert^{(1)}$		$\vee\vert^{(1)}$		$\vee\vert^{(1)}$
BEUP	$\leq^{(2)}$	BVUP	$\leq^{(3)}$	BVDP	$\leq^{(4)}$	BEDP
BEUC	$\leq^{(2)}$	BVUC	$\leq^{(3)}$	BVDC	$\leq^{(4')}$	BEDC

Next we define the notion of parameterized reduction. Let Π_1 and Π_2 be parameterized problems over the alphabets Σ_1 and Σ_2 respectively. We say that Π_1 is (uniformly many-one) FPT-*reducible* to Π_2 if there exist functions $f, g : \mathbb{N} \to \mathbb{N}$, and a mapping $\Phi : \Sigma_1^* \times \mathbb{N} \to (\Sigma_2)^* \times \mathbb{N}$, which is computable in time $f(k)|x|^{O(1)}$ and if $(x', k') = \Phi(x, k)$ then $k' \leq g(k)$, such that $(x, k) \in \Pi_1$ if and only if $\Phi(x, k) \in \Pi_2$. The mapping Φ is called the FPT-*reduction* or *parameterized reduction*. It follows that if a problem Π_1 is in FPT and it can be reduced to Π_2, then Π_2 is in FPT too. To show that a problem is W[1]-hard, it is enough to give a parameterized reduction from a known W[1]-hard problem.

Table 2 illustrates reductions between all considered problems. Here $\Pi_1 \leq^{(i)} \Pi_2$ means that the problem Π_1 can be reduced to the problem Π_2 by the reduction rule i. The edge undirected versions are reduced to the vertex undirected ones by the following rule:

Rule (2): Take the line graph L_G of G and for every clique K of L_G corresponding to the edges incident with a terminal v of G, add a new terminal vertex v' and connect it with all the vertices of the clique. Vertex-disjoint paths of length $l + 1$ in the new graph correspond to edge-disjoint paths of length l in the original graph, while it trivially follows that edge cuts become vertex cuts.

Certainly, vertex undirected versions are reduced to vertex directed ones by the following obvious rule:

Rule (3): Replace every edge by two opposite direction edges.

The following rules reduces all vertex directed versions to their edge directed counterparts. The next rule is given for disjoint path problems:

Rule (4): Replace every vertex v by a directed edge (v_t, v_h) (we call such edges *new edges*) and make v_t the head of all previous edges whose head was v and v_h the tail of all previous edges whose tail was v. For each source vertex s, s_t is a new source, and for each target vertex t, t_h is a new target. Notice that every path of length at most $2l + 1$ between a source and the corresponding target in the new graph corresponds to a path of length at most l in the original graph, and edge-disjoint paths between terminals in the new graph correspond to vertex-disjoint paths in the original graph and vice versa. This proves the correctness of Rule (4) for disjoint path problems.

For cut problems, the rule should be modified:

Rule (4′): Replace every non-terminal vertex v by a directed edge (v_t, v_h) and make v_t the head of all previous edges whose head was v and v_h the tail of all previous edges whose tail was v. Every path of length at most $2l - 1$ between the source and the target in the new graph corresponds to a path of length at most l in the original graph between these terminals. Observe that every vertex cut of the original graph corresponds to an edge cut in the new graph. For the inverse direction, take an edge cut of the new graph and replace each non-new edge e in it with some new edge that has a common non-terminal endpoint with e. This makes every edge cut in the new graph correspond to a vertex cut in the original graph.

Notice that all rules are parameterized reductions when the parameter is k, l, or both.

All problems in Table 1 have two possible parameters k and l in their inputs. Therefore, we consider parameterizations of them with respect to l, k, or both, indicating which parameterization we pick in each problem. For example, the BEUP problem is denoted as BEUP(k) when parameterized by the number of paths k, BEUP(l) when parameterized by the maximum length l of a path and BEUP(k, l) when parameterized by both these quantities. We follow the same notation for all problems in Table 1.

We prove that all variants of our problems are in FPT when parameterized by both k and l. To do it, we give FPT-algorithms for BEDMP(k, l) (Theorem 1, Section 2.1) and BEDC(k, l) (Theorem 2, Section 2.2). Then the claim that all considered problems are in FPT immediately follows from the described reductions between them.

All problems we consider are NP-hard for fixed values of l, bigger than some constant [8,9]. Using standard terminology from [23], this means that all of them, parameterized by l, are para-NP-complete (i.e. they are NP-hard even for fixed values of the parameter). Moreover, the problem asking for the existence of two paths of bounded length between two terminals of a graph is also NP-complete, because of the results in [8,25,26]. This implies the para-NP-completeness of all the disjoint paths variants when parameterized by k. However, no similar result can be expected (unless P = NP) for bounded cut problems, as they trivially admit an $n^{O(k)}$-step algorithm (just check all possible cuts of size at most k). It appears that this running time cannot really become better: we prove that these four variants are W[1]-hard (Theorem 4, Section 3) and that for the directed graph variants, this holds even for directed acyclic graphs (Theorem 3, Section 3). This indicates that, apart from the combinatorial discrepancy between problems on paths and problems on cuts, there is also a discrepancy on the parameterized complexities of the corresponding problems. We stress that this distinction cannot be made clear by studying the classic complexity of the two families of problems (they are all NP-complete in general). Our results are depicted in Table 3.

Our next step is to study the (in general para-NP-complete) parameterized problems BVDP(l) and BVUP(l) for the special case where their input graphs are sparse. We prove (Theorem 7, Section 4.1) that both problems admit FPT-algorithms for

Table 3
Summary of our results when parameterizing by l and k.

	l	k, l	k
BEDMP BVDMP BVUMP BEUMP	para-NP-c [8]	FPT $O(2^{O(kl)} \cdot m \cdot \log n)$ (Theorem 1)	para-NP-c [25,26]
BEDP BVDP BVUP BEUP	para-NP-c [8]	FPT $O(2^{O(kl)} \cdot m \cdot \log n)$ (Theorem 1)	para-NP-c [25,26]
BEDC BVDC BVUC BEUC	para-NP-c [9]	FPT $O(l^{k+1} \cdot m)$ (Theorem 2)	W[1]-h for DAGs (Theorem 3) W[1]-h for DAGs (Theorem 3) W[1]-h (Theorem 4) W[1]-h (Theorem 4)

Table 4
Summary of our results for sparse graph families.

	l, [bounded **ltw**]	l, [two-apex]	**tw**, l
BVUP	FPT (Theorem 7)	para-NP-c (Theorem 8), $l \geq 6$	FPT (Lemma 6)
BVDP	FPT (Theorem 7)	para-NP-c for DAGs (Theorem 8), $l \geq 6$	FPT (Lemma 6)
BEUP	open	para-NP-c (Theorem 10), $l \geq 7$	W[1]-h for fixed $l \geq 10$ (Theorem 11)
BEDP	open	para-NP-c for DAGs (Theorem 10), $l \geq 7$	W[1]-h for DAGs for fixed $l \geq 10$ (Theorem 11)

classes of graphs that have bounded local treewidth (typical graph class with bounded local treewidth are planar graphs or bounded-degree graphs). Moreover, this result can be extended for classes of graphs where the removal of at most one vertex includes them in some bounded local treewidth class.

On the other side, we prove that this sparsity criterion cannot be relaxed much: BVDP(l) (BVUP(l)) remains para-NP-complete (Theorem 8, Section 4.2) for $l \geq 6$, on undirected (directed acyclic) graphs that can be made planar after removing 2 vertices (we call these graphs 2-*apex-graphs*). We also prove that the same holds for the edge variants of the same problems (Theorem 10, Section 4.2) for $l \geq 7$. Our results suggest a rapid change on the problem complexity with respect to the minor-exclusion sparsity criterion.

Our last result concerns the case where BEUP and BEDP are parameterized by the treewidth of their input graphs. We prove that BEUP is W[1]-hard when parameterized by the treewidth of the input graph and that BEDP is W[1]-hard when parameterized by the treewidth of the underlying graph of its input graph even when the input graph is acyclic (Theorem 11, Section 4.3). This last result indicates that the edge-disjoint variants are harder than the vertex-disjoint ones (the same parameterization leads to an FPT-algorithm for BVUP and BVDP— Lemma 6). Our results on sparse graph classes are summarized in Table 4.

2. Parameterized algorithms

2.1. An FPT-algorithm for BEDMP(k, l)

Our algorithm for the BEDMP(k, l) is based on the color-coding technique introduced by Alon et al. in [27]. In particular, we consider a family $\mathcal{F}$ of hash functions, each mapping $\{1, \ldots, m\}$ to a set of colors $\{1, \ldots, k \cdot l\}$, such that for every $S \subseteq \{1, \ldots, m\}$, where $|S| \leq k \cdot l$, there is a $f \in \mathcal{F}$ such that its restriction to S is a bijection. As mentioned in [27], such a family where $|\mathcal{F}| = 2^{O(k \cdot l)} \cdot \log m$ can be constructed in $2^{O(k \cdot l)} \cdot m \cdot \log m$ steps.

Let $\mathcal{F}$ be a family of hash functions as above where $\{1, \ldots, m\}$ represent the edges of G. Let also $\chi \in \mathcal{F}$. Given an integer $i \in \{1, \ldots, k\}$, we define a Boolean function B_i^{χ} such that, for every set of colors $X \subseteq \{1, \ldots, k \cdot l\}$, $B_i^{\chi}(X)$ is true if and only if there exists a collection of i edge-disjoint paths $P_1, \ldots, P_i$ of length at most l such that

- for $j \in \{1, \ldots, i\}$, the endpoints of P_j are s_j and t_j,
- the set of the colors assigned to the edges of these paths is a subset of X (i.e. $\chi(\cup_{j \in \{1,\ldots,i\}} E(P_j)) \subseteq X$),
- each color is used on at most one path.

Notice that an instance of BEDMP(k, l) is a YES-instance if and only if there is a $\chi \in \mathcal{F}$ such that $B_k^{\chi}(\{1, \ldots, k \cdot l\}) = \text{true}$. Let C_i^{χ} be a Boolean function such that, for $X \subseteq \{1, \ldots, k \cdot l\}$, the value of $C_i^{\chi}(X)$ is true if and only if the subgraph of G induced by the edges colored by colors in X contains a path between s_i and t_i of length at most l. Notice that $B_1^{\chi}(X) = C_1^{\chi}(X)$. In general, to compute $B_i^{\chi}(X)$ for some $X \subseteq \{1, \ldots, k \cdot l\}$ and $i > 1$, we observe that

$$B_i^{\chi}(X) = \bigvee_{Y \subseteq X} (B_{i-1}^{\chi}(Y) \wedge C_i^{\chi}(X \setminus Y)).$$

The value of $C_i^X(X)$ for a given set X can be computed in $O(m)$ steps. Therefore, computing $B_i^X(X)$ for all $X \subseteq \{1, \dots, l \cdot k\}$ requires $O(3^{k \cdot l} \cdot m)$ steps once the values of B_{i-1}^X are known. Hence the above dynamic programming requires in total $O(3^{k \cdot l} \cdot m \cdot k)$ steps to compute $B_k^X(\{1, \dots, k \cdot l\})$. Concluding, BEDMP$(k, l)$ can be solved in $O(2^{O(k \cdot l)} \log m \cdot m \cdot k)$ steps.

It is easy to observe that the above algorithm can be modified so that it also would return the requested paths when they exist. We conclude with the following.

Theorem 1. *The* BEDMP(k, l) *problem can be solved by an* FPT*-algorithm that runs in* $O(2^{O(k \cdot l)} \log n \cdot m \cdot k)$ *steps where* $n = |V(G)|$ *and* $m = |E(G)|$.

2.2. An FPT-algorithm for BEDC(k, l)

Our algorithm for the BEDC problem is called **Solve-BEDC**. The algorithm is based on the simple observation that, for any (s, t)-path of length at most l, at least one edge of it has to be included in any (s, t)-edge l-bounded cut. The algorithm's input contains the input of the problem and a set $X \subseteq E(G)$. The algorithm returns either an (s, t)-edge l-bounded cut S, such that $X \subseteq S$, or returns the answer **NO** if such a cut does not exist. To solve BEDC for a graph G with terminal vertices s and t, it is enough to call **Solve-BEDC** $(G, s, t, k, l, \emptyset)$.

Algorithm 1 Solve-BEDC(G, s, t, k, l, X).

Input: A graph G, vertices $s, t \in V(G)$, k, l and a set $X \subseteq E(G)$.
Output: An (s, t)-edge bounded cut $S \supseteq X$ or **NO** if such a cut does not exist.

if $|X| > k$ **then return NO**;
let P be a shortest (s, t)-path in $G \setminus X$;
if $|E(P)| > l$ **then return** X
  otherwise
  for every edge $e \in E(P)$
    set $Y =$ **Solve-BEDMP**$(G, s, t, k, l, X \cup \{e\})$;
    if $Y \neq$ **NO then return** Y
  return NO.

The correctness of the algorithm directly follows from the description. Clearly, if $|X| > k$ then the answer is **NO**. Then the algorithm finds the shortest (s, t)-path in $G \setminus X$. If the path has a length bigger than l (or does not exist) then X is an l-bounded cut. Otherwise, at least one edge of the path should be included in any l-bounded cut. We branch by considering at most l possible choices of an edge. Since at each recursive call of **Solve-BEDC**, either the cardinality of the parameter X is increased by one or we stop, the depth of the search tree is $k+1$ at most. The number of branches at each call of **Solve-BEDC** is at most l. So the number of calls is at most $O(l^{k+1})$. Since we can find the shortest path between two vertices in $O(m)$ for a graph with m edges, the total running time is $O(l^{k+1} \cdot m)$. This yields the following result.

Theorem 2. *The* BEDC(k, l) *problem can be solved by an* FPT*-algorithm that runs in* $O(l^{k+1} \cdot m)$ *time where* $m = |E(G)|$.

3. Hardness results for (s, t)-cuts

In this section we prove W[1]-hardness of BVDC(k) and BEUC(k). It can be noted that by the reduction rules (see Table 2) W[1]-hardness of BVDC(k) follows from a similar result for BEUC(k), but we prove here a stronger result.

Theorem 3. *The* BVDC(k) *problem is* W[1]*-hard even for acyclic directed graphs.*

Proof. We present a reduction from the Multicolored Clique problem, which is defined as follows:

Multicolored clique

Input: A graph G with a proper k-coloring of G.
Question: Is there a clique of size k in G containing exactly one vertex from each color class?

The Multicolored Clique problem, parameterized by k, was proved to be W[1]-hard by Fellows et al. [28].

Let G be an n-vertex undirected graph. Denote by X_i the i-th color class in the given k-coloring of G. Assume, without loss of a generality, that $k \geq 4$. We assume also that for any pair of sets X_i, X_j, $i \neq j$, vertices of these sets are connected by the same number of edges denoted by m, and $m > 0$ (otherwise it is possible to add pairs of adjacent vertices to the graph to ensure this condition). Denote by $e_1^{(i,j)}, e_2^{(i,j)}, \dots, e_m^{(i,j)}$ the edges which join sets X_i and X_j. Let $l = 5m + 4$.

Now we consider auxiliary constructions. For every $i, j \in \{1, 2, \dots, k\}$, $i \neq j$, a directed graph $F_{i,j}$ is constructed as follows (the graph $F_{i,j}$ is shown in Fig. 1 for $m = 4$).

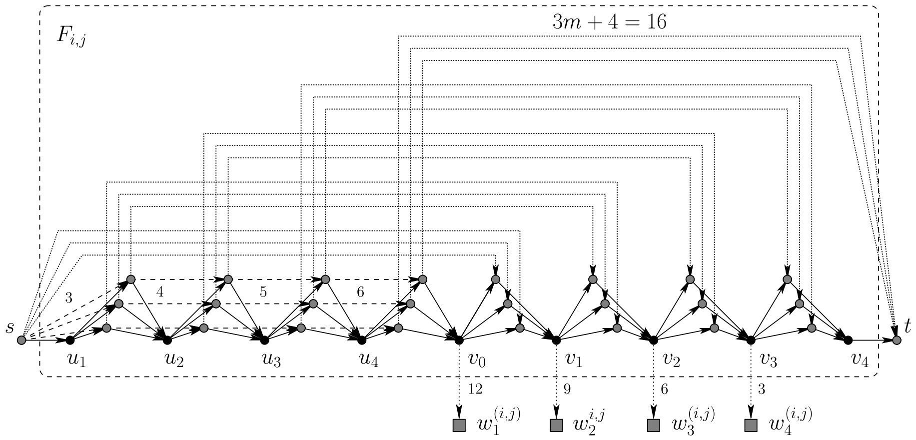


Fig. 1. Construction of $F_{i,j}$ for $m = 4$. Paths are shown by dashed lines.

1. Two vertices s and t are created.
2. For every $r \in \{1, 2, \ldots, m\}$, vertices $u_r, a_r^{(1)}, a_r^{(2)}, a_r^{(3)}$ and $b_r^{(1)}, b_r^{(2)}, b_r^{(3)}$ are constructed, and for every $r \in \{0, 1, \ldots, m\}$, a vertex v_r is introduced. It is assumed, for convenience, that $s = a_0^{(1)} = a_0^{(2)} = a_0^{(3)}, b_0^{(1)} = a_m^{(1)}, b_0^{(2)} = a_m^{(2)}, b_0^{(3)} = a_m^{(3)}$ and $t = b_{m+1}^{(3)} = b_{m+1}^{(3)} = b_{m+1}^{(3)}$.
3. For each vertex u_r, edges $(a_{r-1}^{(1)}, u_r), (a_{r-1}^{(2)}, u_r), (a_{r-1}^{(3)}, u_r)$ and $(u_r, a_r^{(1)}), (u_r, a_r^{(2)}), (u_r, a_r^{(3)})$ are added.
4. For each vertex v_r, edges $(b_r^{(1)}, v_r), (b_r^{(2)}, v_r), (b_r^{(3)}, v_r)$ and $(v_r, b_{r+1}^{(1)}), (v_r, b_{r+1}^{(2)}), (v_r, b_{r+1}^{(3)})$ are added.
5. Pairs of vertices $a_{r-1}^{(f)}, a_r^{(f)}$ are joined by paths of length $r + 2$ for $f = 1, 2, 3$ and $r \in \{1, 2, \ldots, m\}$.
6. Pairs of vertices $a_{r-1}^{(f)}, b_r^{(f)}, f = 1, 2, 3$, are joined by paths of length $3m + 4$ for $r \in \{1, 2, \ldots, m + 1\}$.
7. Add vertices $w_1^{(i,j)}, w_2^{(i,j)}, \ldots, w_m^{(i,j)}$, and join every vertex v_{r-1} with $w_r^{(i,j)}$ by a path of length $3(m + 1 - r)$.

Some properties of this graph are given in the next claim.

Claim 1. *For $i, j \in \{1, \ldots, k\}$, $i \neq j$, the following holds:*

(a) *any (s, t)-vertex l-bounded cut in $F_{i,j}$ contains at least two vertices;*
(b) *the set $X_r^{(i,j)} = \{u_r, v_r\}$ is an (s, t)-vertex l-bounded cut in $F_{i,j}$ such that the vertex $w_r^{(i,j)}$ is joined with s by a path of length at most $l - 2$ in $F_{i,j} \setminus X$, and other vertices $w_f^{(i,j)}$ can be joined with s only by paths of length at least $l - 1$;*
(c) *for any two vertex (s, t)-vertex l-bounded cut X in $F_{i,j}$, at least one vertex $w_r^{(i,j)}$ is joined with s by a path of length at most $l - 2$ in $F_{i,j} \setminus X$.*

Proof of Claim 1. Part (a) is proved by checking directly that a single vertex cannot be an (s, t)-vertex l-bounded cut. Clearly, if we remove a vertex $x \notin \{u_1, \ldots, u_m\} \cup \{v_0, \ldots, v_m\}$ then the obtained graph still has an (s, t)-path of length $4m+2 < 5m+4 = l$. For a vertex $x \in \{u_1, \ldots, u_m\} \cup \{v_0, \ldots, v_m\}$, there is an (s, t)-path of length $5m+4 = l$ in $F_{i,j} \setminus \{x\}$. To prove (b) and (c), it is sufficient to note that by similar arguments as for (a), if X is a two element (s, t)-vertex cut in $F_{i,j}$ then $X = \{u_f, v_h\}$ for some $f, h \in \{1, 2, \ldots, m\}$ and $f \leq h$. It remains to observe that for such a set X, vertices $w_f^{(i,j)}, \ldots, w_h^{(i,j)}$ are joined with s by paths of length at most $l - 2$ in $F_{i,j} \setminus X$, and other vertices $w_l^{(i,j)}$ can be joined with s only by paths of length at least $l - 1$ in $F_{i,j} \setminus X$. □

Using these gadgets $F_{i,j}$, we construct a directed graph H from G as follows.

8. For all pairs $\{i, j\}$, $i, j \in \{1, \ldots, k\}$, $i \neq j$, graphs $F_{i,j}$ with common vertices s and t are constructed.
9. Add all vertices of G to H.
10. For every edge $e_f^{(i,j)} = \{x, y\}$ of G, we add the two directed edges $(w_f^{(i,j)}, x)$ and $(w_f^{(i,j)}, y)$ to H.
11. For each vertex $x \in V(G)$, an edge (x, t) is added to H.

It is easy to see that H is a directed acyclic graph. The next claim concludes the proof of Theorem 3.

Claim 2. *Graph G has a clique of size k which contains exactly one vertex from each color class if and only if there is an (s, t)-vertex l-bounded cut in H with at most $k' = k^2$ vertices.*

Proof of Claim 2. Suppose that C is a clique in G of size k which contains exactly one vertex from each color class. We construct the (s, t)-vertex l-bounded cut X in H as follows. All vertices of C are included in X. For every edge $e_f^{(i,j)}$ which joins vertices of C in G, the vertices of $X_f^{(i,j)}$ from $F_{i,j}$ are included in X. It follows immediately from Claim 1(b) that X is an (s, t)-vertex l-bounded cut in H. Clearly, $|X| = k + 2\frac{k(k-1)}{2} = k'$.

Assume now that X is an (s, t)-vertex l-bounded cut in H of size at most k'. By Claim 1(a), at least two vertices of X belong to every graph $F_{i,j}$. Suppose that for some color class X_i, $X_i \cap X = \emptyset$. Then by Claim 1(c), the set X must contain at least three vertices from graphs $F_{i,j}$ for $j \in \{1, 2, \ldots, k\}, j \neq i$. Therefore there is another color class X_r such that $X_r \cap X = \emptyset$, and X contains at least three vertices from graphs $F_{r,j}$ for $j \in \{1, 2, \ldots, k\}, j \neq r$. But in this case X has at least $2\frac{k(k-1)}{2} + 2k - 3 = k^2 + k - 3$ vertices, and this means that $k^2 + k - 3 > k'$, since $k \geq 4$. So, every set X_i has at least one element of X. It follows that $|X| = k'$ and X contains exactly two vertices from every graph $F_{i,j}$ and exactly one vertex from every set X_i. Using Claim 1(c) we conclude that $X \cap V(G)$ is a clique in G of size k. Claim 2 follows. □

Notice that the reduction 4 from BVDC(k) to BEDC(k) transforms a directed acyclic graph into another directed acyclic graph. So, W[1]-hardness of BEDC(k) for directed acyclic graphs follows immediately. What remains to prove is the W[1]-hardness for the undirected case.

Theorem 4. BEUC(k) *is* W[1]-*hard.*

Proof. The proof of this theorem uses the same ideas as the proof of Theorem 3. The reduction is almost the same, we describe only the modifications of it. Now we let $l = 7m + 5$. All other changes mainly concern the gadgets $F_{i,j}$:

1. Construct graph $F_{i,j}$ as in the proof of Theorem 3.
2. Replace $(a_{r-1}^{(f)}, a_r^{(f)})$-paths of length $r + 2$ by paths of length $r + 3$.
3. Replace $(a_{r-1}^{(f)}, b_r^{(f)})$-paths of length $3m + 4$ by paths of length $4m + 5$.
4. Replace $(v_{r-1}, w_r^{(i,j)})$-paths of length $3(m + 1 - r)$ by paths of length $4(m + 1 - r)$.
5. For each $r \in \{1, \ldots, m\}$, the vertex u_r is "split" into two vertices $u_r^{(1)}, u_r^{(2)}$: replace u_r by $u_r^{(1)}, u_r^{(2)}$, add the edge $(u_r^{(1)}, u_r^{(2)})$, replace any edge heading in u_r by an edge heading in $u_r^{(1)}$ and each edge tailing in u_r by an edge tailing in $u_r^{(2)}$.
6. Similarly, for each $r \in \{0, \ldots, m\}$, the vertex v_r is "split" into two vertices $v_r^{(1)}, v_r^{(2)}$: replace v_r by $v_r^{(1)}, v_r^{(2)}$, add the edge $(v_r^{(1)}, v_r^{(2)})$, replace any edge heading in v_r by an edge heading in $v_r^{(1)}$ and each edge tailing in v_r by an edge tailing in $v_r^{(2)}$.
7. Replace all directed edges by undirected ones.

This graph has the properties summarized in the next lemma. We omit the proof as it is similar to the first claim of Theorem 3.

Lemma 5. *For $i, j \in \{1, \ldots, k\}, i \neq j$, the following holds:*

(a) *Any (s, t)-edge l-bounded cut in $F_{i,j}$ contains at least two edges.*

(b) *For any two-edge (s, t)-edge l-bounded cut Z in $F_{i,j}$, at least one vertex $w_r^{(i,j)}$ is joined with s by a path of length at most $l - 2$ in $F_{i,j} \setminus Z$.*

(c) *The set $Z_r^{(i,j)} = \{\{u_r^{(1)}, u_r^{(2)}\}, \{v_r^{(1)} v_r^{(2)}\}\}$ is an (s, t)-edge l-bounded cut in $F_{i,j}$ such that the vertex $w_r^{(i,j)}$ is joined with s by a path of length at most $l - 2$ in $F_{i,j} \setminus Z_r^{(i,j)}$, and all other vertices $w_f^{(i,j)}$ can be joined with s only by paths of length at least $l - 1$.*

Using these gadgets $F_{i,j}$ we construct a graph H from G as in the proof of Theorem 3 with the only difference that undirected edges are used instead of directed ones.

The final claim is that the graph G has a clique of size k which contains exactly one vertex from any color class if and only if there is an (s, t)-edge l-bounded cut in H with at most $k' = k^2$ vertices. This claim is proved in the same way as the second claim of Theorem 3 and concludes the proof of Theorem 4. □

4. (s, t)-paths of bounded length for sparse graphs

4.1. FPT-algorithms for sparse graph classes

A *tree decomposition* of a graph G is a pair (X, T) where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a collection of subsets (called *bags*) of $V(G)$ such that:

1. $\bigcup_{i \in V(T)} X_i = V(G)$,
2. for each edge $\{x, y\} \in E(G)$, $x, y \in X_i$ for some $i \in V(T)$, and
3. for each $x \in V(G)$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is $\max_{i \in V(T)} \{|X_i| - 1\}$. The *treewidth* of a graph G (denoted as $\mathbf{tw}(G)$) is the minimum width over all tree decompositions of G. For a directed graph G, $\mathbf{tw}(G)$ is the treewidth of the underlying undirected graph.

We say that a graph class $\mathcal{G}$ has *bounded local treewidth with bounding function* f if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph $G \in \mathcal{G}$, every $v \in V(G)$, and every positive integer i it holds that $\mathbf{tw}(G[N_G^i[v]]) \leq f(i)$ where $N_G^i[v] = \{u \in V(G) \colon \mathrm{dist}_G(u, v) \leq i\}$.

It appears that many sparse graph classes have bounded local treewidth. Examples are planar graphs and graphs of bounded genus, bounded max-degree graphs, and graphs excluding an apex graph as a minor (an apex graph is a graph that can become planar after removal of one vertex). The purpose of this subsection is to construct an FPT algorithm for the BVDP(l) and the BVUP(l) problems when their inputs are restricted to graphs that belong to some (almost) bounded local treewidth graph class. For this, we need the following lemma, which follows directly from the results of Arnborg et al. in [29] (see also [30]).

Lemma 6. *The* BVDP *problem (and therefore, also* BVUP*) parameterized by l and the treewidth of the input graph can be solved by an* FPT*-algorithm.*

We are now ready to prove the following.

Theorem 7. *The* BVDP(l) *problem (and therefore, also* BVUP(l)*) can be solved by an* FPT*-algorithm for graph classes that have bounded local treewidth. Moreover, let $\mathcal{G}$ be a bounded local treewidth graph class, and let $\mathcal{G}'$ be a set of all graphs G such that there is a set $X \subseteq V(G)$, $|X| \leq 1$, for which $G \setminus X \in \mathcal{G}$. Then the* BVDP($l$) *can be solved by an* FPT*-algorithm in $\mathcal{G}'$.*

Proof. Let $\mathcal{G}$ be a graph class with bounded local treewidth and let $G \in \mathcal{G}$. Notice that any vertex of a path of a possible solution of the BVDP(l) problem is in distance at most l from s (or t). We set $G' = G[N_G^l(s)]$ and we observe that the instances (G, k, l, s, t) and (G', k, l, s, t) are equivalent for the BVDP(l) problem. As $\mathcal{G}$ has a bounded local treewidth, we conclude that $\mathbf{tw}(G') \leq f(l)$. The result follows from Lemma 6.

Let now $G \in \mathcal{G}'$. If $X = \emptyset$ then we can use the first claim of the theorem. Suppose that $X = \{u\}$ and let $H = G \setminus X$. If $u = s$ ($u = t$ resp.) then we set $G' = G[N_H^l(t) \cup \{u\}]$ ($G' = G[N_H^l(s) \cup \{u\}]$ resp.) and observe that the instances (G, k, l, s, t) and (G', k, l, s, t) are equivalent for the BVDP(l) problem. Clearly, $\mathbf{tw}(G') \leq f(l) + 1$, and the result follows from Lemma 6, Section 4.1. If $u \neq s, t$ then let $G' = G[N_H^l(s) \cup N_H^l(t) \cup \{u\}]$. Since for any (s, t)-path P in G, all vertices of $P \setminus \{u\}$ are at distance at most l from s or t in H, the instances (G, k, l, s, t) and (G', k, l, s, t) are equivalent. It remains to note that graphs $G[N_H^l(s)]$ and $G[N_H^l(t)]$ are either disjoint or $G[N_H^l(s) \cup N_H^l(t) \cup \{u\}]$ has diameter at most $4l$. Hence G' has bounded treewidth and we can again use Lemma 6. □

4.2. Vertex-disjoint (s, t)-paths of bounded length for H-minor-free graphs

In this section we show that the restrictions of Theorem 7 are somehow tight. We call a graph G a *two-apex graph* if there is a set X of at most two vertices such that $G \setminus X$ is a planar graph.

Theorem 8. *For any fixed $l \geq 6$,*

(a) BVUP(l) *is* NP*-complete for two-apex graphs,*
(b) BVDP(l) *is* NP*-complete for directed acyclic two-apex graphs.*

Proof. First, we prove Part (a). We reduce a variant of the 3-Satisfiability problem which was considered by Kratochvíl [31]. Let C be a Boolean formula ϕ with variables $x_1, x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$ such that each clause contains at most 3 literals. Suppose that H is a bipartite graph with vertices $x_1, x_2, \ldots, x_n$ and $C_1, C_2, \ldots, C_m$ such that x_i and C_j are adjacent if and only if clause C_j contains literal x_i or $\overline{x}_i$. It is known (see [31]) that the 3-Satisfiability problem remains NP-complete even if H is a planar graph and every variable occurs in no more than four clauses. We need here one additional condition.

Lemma 9. *The* 3-Satisfiability *problem remains NP-complete if H has a plane embedding such that for each x_i, if $\deg_H(x_i) = 4$ and x_i occurs in exactly two clauses C_j, C_r in a positive form then $\{x_i, C_j\}$ and $\{x_i, C_r\}$ are adjacent edges in the boundary of one face.*

Proof. Let us consider some planar embedding of H. Suppose that our condition is not fulfilled for some vertex x_i. Then the variable x_i is used in four clauses. Assume that clauses C_j, C_r contain x_i and $C_{j'}, C_{r'}$ contain $\overline{x}_i$. We construct another instance of 3-Satisfiability. We add a new variable x_i' and replace x_i by x_i' in C_r and $C_{r'}$. Then clauses $x_i \vee \overline{x'}_i$ and $\overline{x}_i \vee x_i'$ are added to ϕ. Since $(x_i \vee \overline{x'}_i) \wedge (\overline{x}_i \vee x_i') = \mathtt{true}$ if and only if x_i and x_i' have the same values, these two instances of 3-Satisfiability are equivalent. It remains to observe that for the new instance, the embedding of H can be replaced by an embedding, for which our condition for x_i and x_i' is fulfilled. □

Notice that if this condition holds, then it is possible to "split" every vertex x_i in H into two adjacent vertices x_i and $\overline{x}_i$ in a way that

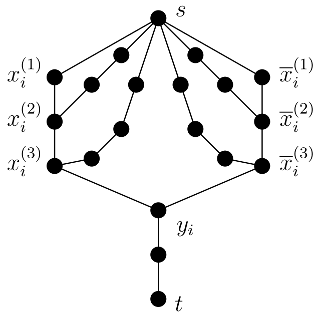
- x_i is adjacent to C_j if the clause C_j contains literal x_i, and
- $\overline{x}_i$ is adjacent to C_j if the clause C_j contains literal $\overline{x}_i$,

in such a way, that after this splitting the graph remains planar.

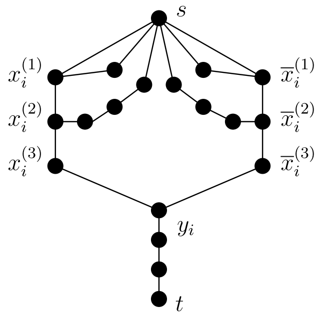
We also assume that no variable occurs only in positive or only in negations in all clauses.

For every $i \in \{1, 2, \ldots, n\}$ we define a graph F_i as follows (the graph F_i for $l = 6$ is shown in Fig. 2(a)).

1. Introduce three vertices s, t and y_i.
2. Add vertices $x_i^{(1)}, x_i^{(2)}, x_i^{(3)}$ and edges $\{s, x_i^{(1)}\}$, $\{x_i^{(1)}, x_i^{(2)}\}$, $\{x_i^{(2)}, x_i^{(3)}\}$, $\{x_i^{(3)}, y_i\}$.
3. Add vertices $\overline{x}_i^{(1)}, \overline{x}_i^{(2)}, \overline{x}_i^{(3)}$ and edges $\{s, \overline{x}_i^{(1)}\}$, $\{\overline{x}_i^{(1)}, \overline{x}_i^{(2)}\}$, $\{\overline{x}_i^{(2)}, \overline{x}_i^{(3)}\}$, $\{\overline{x}_i^{(3)}, y_i\}$.



(a) BVUP(l), $l = 6$.



(b) BEUP(l), $l = 7$.

Fig. 2. Graphs F_i.

4. Join vertices y_i and t by a path P_i of length $l - 4$.
5. Join s with $x_i^{(2)}$ and $\overline{x}_i^{(2)}$ by paths $Q_i^{(2)}$ and $\overline{Q}_i^{(2)}$ of length 3.
6. Join s with $x_i^{(3)}$ and $\overline{x}_i^{(3)}$ by paths $Q_i^{(3)}$ and $\overline{Q}_i^{(3)}$ of length 4.

Let $Q_i^{(1)} = \langle s, x_i^{(1)}\rangle$ and $\overline{Q}_i^{(1)} = \langle s, \overline{x}_i^{(1)}\rangle$.

We complete the reduction as follows.

7. For each variable x_i, a gadget F_i with common vertices s and t is constructed.
8. Vertices $C_1, C_2, \dots, C_m$ are added.
9. Assume that for a variable x_i, clauses $C_{i_1}, \dots, C_{i_{p_i}}$ contain the literal x_i and clauses $C_{i'_1}, \dots, C_{i'_{q_i}}$ contain $\overline{x}_i$. Notice $p_i \le 3$ and $q_i \le 3$. The vertices $\{x_i^{(1)}, \dots, x_i^{(p_1)}\}$ are joined by a matching with the vertices $C_{i_1}, \dots, C_{i_{p_i}}$ and the vertices $\overline{x}_i^{(1)}, \dots, \overline{x}_i^{(q_i)}$ are joined with the vertices $C_{i'_1}, \dots, C_{i'_{q_i}}$.
10. Vertices C_j are joined by edges with t.

Denote by H the obtained graph. By Lemma 9 H is a two-apex graph with apices s and t. We claim that there is a truth assignment for variables $x_1, x_2, \dots, x_n$ such that the Boolean formula is satisfied if and only if H has at least $n + m$ vertex-disjoint (s, t)-paths of length at most l.

Suppose that variables $x_1, x_2, \dots, x_n$ have a truth assignment which satisfies the formula. For every $i \in \{1, 2, \dots, n\}$, we chose the path $\langle s, x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, y_i\rangle \oplus P_i$ if $x_i = \mathsf{false}$ and the path $\langle s, \overline{x}_i^{(1)}, \overline{x}_i^{(2)}, \overline{x}_i^{(3)}, y_i\rangle \oplus P_i$ if $x_i = \mathsf{true}$. Every clause C_j contains a literal z that is evaluated true. If $z = x_i$ then H has an edge $\{x_i^{(r)}, C_j\}$ for some $r = 1, 2, 3$, and we chose the path $Q_i^{(r)} \oplus \langle x_i^{(r)}, C_j, t\rangle$. If $z = \overline{x}_i$ then H has an edge $\{\overline{x}_i^{(r)}, C_j\}$ and the path $\overline{Q}_i^{(r)} \oplus \langle \overline{x}_i^{(r)}, C_j, t\rangle$ is considered. Clearly, all these $n + m(s, t)$-paths are vertex-disjoint and have length at most l.

Assume now that H has a collection of at least $n+m$ vertex-disjoint (s, t)-paths of length at most l. Since $\deg_H(t) = n+m$, all vertices $y_1, y_2, \dots, y_n$ and all vertices $C_1, C_2, \dots, C_m$ belong to different paths in this collection. The path which contains y_i is either $\langle s, x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, y_i\rangle \oplus P_i$ or $\langle s, \overline{x}_i^{(1)}, \overline{x}_i^{(2)}, \overline{x}_i^{(3)}, y_i\rangle \oplus P_i$ because the length of P_i is $l - 4$. We set $x_i = \mathsf{false}$ if this path goes through $x_i^{(1)}, x_i^{(2)}, x_i^{(3)}$, and x_i is set true otherwise. Every path which contains C_j has an edge $\{C_j, z\}$ for some vertex z from some graph F_i. If $x_i = \mathsf{true}$ by our assignment, then $\overline{x}_i^{(1)}, \overline{x}_i^{(2)}, \overline{x}_i^{(3)}$ are included in the path which contains y_i. Hence $z \in \{x_i^{(1)}, x_i^{(2)}, x_i^{(3)}\}$. Similarly, if x_i is set false, then $z \in \{\overline{x}_i^{(1)}, \overline{x}_i^{(2)}, \overline{x}_i^{(3)}\}$. It means that every clause is satisfied by our truth assignment.

The Part (b) of the theorem is proved by almost the same reduction. The only difference is that we have to replace graph H by a directed acyclic graph which is obtained by directing all edges "from s to t". □

Consider the class of K_r-minor free graphs (i.e. none of the graphs in this class contains a subgraph that can be contracted to K_r). Notice that K_5-free graphs have bounded local treewidth. However this is not true for K_r-minor-free graphs for $r \ge 6$. Since two-apex graphs are K_7-minor-free, Theorem 7 provides a nearly optimal estimation on the tractability of BVUP(l) and BVDP(l) on K_r-minor-free graphs. Actually, the same theorem argues that not even an $n^{f(k)}$ step algorithm can be found for $r \ge 7$ unless P = NP.

We also prove the following theorem.

Theorem 10. *For any fixed $l \ge 7$,*

(a) BEUP(l) *is NP-complete for two-apex graphs,*
(b) BEDP(l) *is NP-complete for directed acyclic two-apex graphs.*

Proof. The proof uses the same ideas as the proof of Theorem 8. So, we describe here only the differences in constructions of F_i and H for this case (the graph F_i for $l = 7$ is shown in Fig. 2(b)):

5. Join s with $x_i^{(1)}$ and $\bar{x}_i^{(1)}$ by paths $Q_i^{(2)}$ and $\overline{Q}_i^{(2)}$ of length 2.
6. Join s with $x_i^{(2)}$ and $\bar{x}_i^{(2)}$ by paths $Q_i^{(3)}$ and $\overline{Q}_i^{(3)}$ of length 4.
9. Let $C_{i_1}, \dots, C_{i_{p_i}}$ be clauses which contain x_i (note that $p_i \le 3$). Join $x_i^{(j)}$ and C_{i_j} by a path of length $l - 2j$ for $j \in \{1, \dots, p_i\}$. Similarly, let now $C_{i'_1}, \dots, C_{i'_{q_i}}$ be clauses which contain $\bar{x}_i$. Join $\bar{x}_i^{(j)}$ and $C_{i'_j}$ by a path of length $l - 2j$ for $j \in \{1, \dots, q_i\}$.

The second part of the theorem is proved by the same reduction by replacing graph H by a directed acyclic graph which is obtained by directing edges of H in the obvious way. □

4.3. Edge-disjoint (s, t)-paths of bounded length for graphs of bounded treewidth

By reduction rules (see Table 2) BEUP can be reduced to BVUP, but Reduction rule (2) does not preserve the treewidth of the graph. The following theorem makes a contrast to Lemma 6: it shows that edge-disjoint path problems are harder than their vertex-disjoint counterparts, when parameterized by the treewidth.

Theorem 11. *For every fixed $l \ge 10$,*

(a) BEUP *is* W[1]*-hard when parameterized by treewidth, and*
(b) BEDP *is* W[1]*-hard for directed acyclic graphs when parameterized by the treewidth of the underlying graph.*

Proof. The parameterized complexity of the Capacitated Dominating Set was considered in [32], and it was proved that this problem is W[1]-hard when parameterized by both the size of the capacitated dominating set and the treewidth of the input graph. Here we reduce a special variant of this problem. A *red–blue capacitated graph* is a pair (G, c), where G is a bipartite graph with the vertex bipartition R and B and $c: R \to \mathbb{N}$ is a *capacity* function such that $1 \le c(v) \le d_G(v)$ for every vertex $v \in R$. The vertices of the set R are called *red* and the vertices of B are called *blue*. A set $S \subseteq R$ is called a *capacitated dominating set* if there is a *domination mapping* $f: B \to S$ which maps every vertex in B to one of its neighbors such that the total number of vertices mapped by f to any vertex $v \in S$ does not exceed its capacity $c(v)$. The Red–Blue Capacitated Dominating Set problem is formulated as follows:

Red–Blue Capacitated Dominating Set

Input: A red–blue capacitated graph (G, c) and a positive integer k.
Question: Does there exist a capacitated dominating set S for G containing at most k vertices?

It follows from the results of [33] (see [34]) that the Red–Blue Capacitated Dominating Set problem is W[1]-hard when parameterized by the treewidth t of the input graph and the solution size k.

At first we prove W[1] hardness for the BEUP. To simplify the reduction we describe it for $l = 10$. To extend the proof for $l > 10$, it is enough to replace all edges incident with t in the graph constructed here by paths of length $l - 9$. Let R and B sets of red and blue vertices of G respectively. Let also $n = |R|$, $m = |B|$ and $r = \max_{v \in R} c(v)$. We start the description of our reduction with some auxiliary gadgets.

For $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, r\}$, we define the graph $F_i^{(j)}$ as follows.

1. Introduce vertices $s, x^{(i,j)}, y^{(i,j)}, z^{(i,j)}, t$, and add edges $\{s, x^{(i,j)}\}$, $\{y^{(i,j)}, z^{(i,j)}\}$ and $\{z^{(i,j)}, t\}$.
2. Add vertices $a_1^{(i,j)}, a_2^{(i,j)}, c_1^{(i,j)}, c_2^{(i,j)}, e_1^{(i,j)}, e_2^{(i,j)}$, and edges $\{x^{(i,j)}, a_1^{(i,j)}\}$, $\{a_1^{(i,j)}, a_2^{(i,j)}\}$, $\{a_2^{(i,j)}, c_1^{(i,j)}\}$, $\{c_1^{(i,j)}, c_2^{(i,j)}\}$, $\{c_2^{(i,j)}, e_1^{(i,j)}\}$, $\{e_1^{(i,j)}, e_2^{(i,j)}\}$ and $\{e_2^{(i,j)}, y^{(i,j)}\}$. We call the path $\langle s, x^{(i,j)}, a_1^{(i,j)}, a_2^{(i,j)}, c_1^{(i,j)}, c_2^{(i,j)}, e_1^{(i,j)}, e_2^{(i,j)}, y^{(i,j)}, z^{(i,j)}, t\rangle$ the *lower* path for $F_i^{(j)}$.
3. Add vertices $b_1^{(i,j)}, b_2^{(i,j)}, d_1^{(i,j)}, d_2^{(i,j)}, f_1^{(i,j)}, f_2^{(i,j)}$, and edges $\{x^{(i,j)}, b_1^{(i,j)}\}$, $\{b_1^{(i,j)}, b_2^{(i,j)}\}$, $\{b_2^{(i,j)}, d_1^{(i,j)}\}$, $\{d_1^{(i,j)}, d_2^{(i,j)}\}$, $\{d_2^{(i,j)}, f_1^{(i,j)}\}$, $\{f_1^{(i,j)}, f_2^{(i,j)}\}$ and $\{f_2^{(i,j)}, y^{(i,j)}\}$. We call the path $\langle s, x^{(i,j)}, b_1^{(i,j)}, b_2^{(i,j)}, d_1^{(i,j)}, d_2^{(i,j)}, f_1^{(i,j)}, f_2^{(i,j)}, y^{(i,j)}, z^{(i,j)}, t\rangle$ the *upper* path for $F_i^{(j)}$.

The next step is the construction of graphs F_i for $i \in \{1, \dots, n\}$ (see Fig. 3).

4. Construct graphs $F_i^{(1)}, \dots, F_i^{(r)}$ with common vertices s and t.
5. – For each $j \in \{1, \dots, \lfloor r/2 \rfloor\}$, add two vertices $g^{(i,2j-1)}, h^{(i,2j-1)}$ and edges $\{s, g^{(i,2j-1)}\}$, $\{h^{(i,2j-1)}, t\}$.
   – Join $g^{(i,2j-1)}$ and $a_1^{(i,2j-1)}$ by a path of length 2, add an edge $\{a_2^{(i,2j-1)}, a_1^{(i,2j)}\}$, and join $a_2^{(i,2j)}$ with $h^{(i,2j-1)}$ by a path of length 3. Denote the unique (s, t)-path which contains the two paths constructed here and the edges $\{s, g^{(i,2j-1)}\}$, $\{a_1^{(i,2j-1)}, a_2^{(i,2j-1)}\}$, $\{a_2^{(i,2j-1)}, a_1^{(i,2j)}\}$, $\{a_1^{(i,2j)}, a_2^{(i,2j)}\}$, and $\{h^{(i,2j-1)}, t\}$ by $P_l^{(i,2j-1)}$.
   – Join $g^{(i,2j-1)}$ and $b_1^{(i,2j-1)}$ by a path of length 2, add an edge $\{b_2^{(i,2j-1)}, b_1^{(i,2j)}\}$, and join $b_2^{(i,2j)}$ with $h^{(i,2j-1)}$ by a path of length 3. Denote the unique (s, t)-path which contains the two paths constructed here and the edges $\{s, g^{(i,2j-1)}\}$, $\{b_1^{(i,2j-1)}, b_2^{(i,2j-1)}\}$, $\{b_2^{(i,2j-1)}, b_1^{(i,2j)}\}$, $\{b_1^{(i,2j)}, b_2^{(i,2j)}\}$, and $\{h^{(i,2j-1)}, t\}$ by $P_u^{(i,2j-1)}$.
6. – For each $j \in \{1, \dots, \lfloor (r-1)/2 \rfloor\}$, add two vertices $g^{(i,2j)}, h^{(i,2j)}$ and edges $\{s, g^{(i,2j)}\}$, $\{h^{(i,2j)}, t\}$.

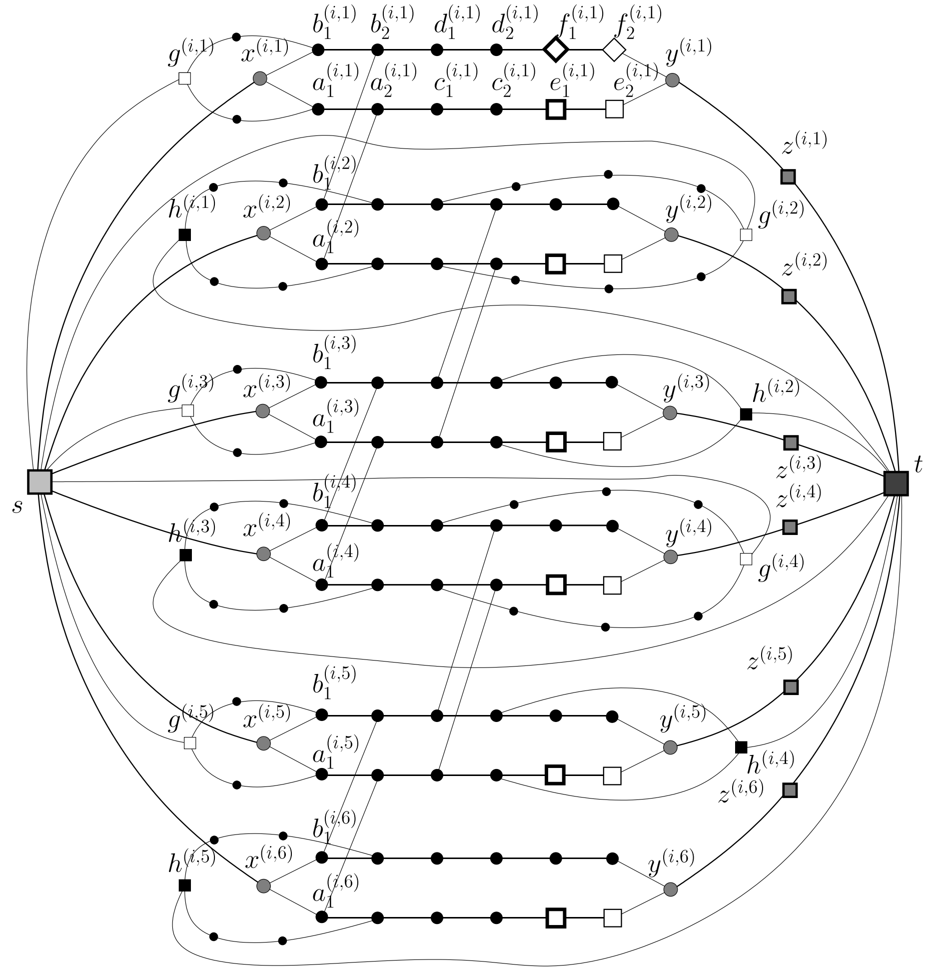


Fig. 3. The construction of F_i for $r = 6$.

– Join $g^{(i,2j)}$ and $c_1^{(i,2j)}$ by a path of length 4 and add edges $\{c_2^{(i,2j)}, c_1^{(i,2j+1)}\}$, $\{c_2^{(i,2j+1)}, h^{(i,2j)}\}$. Denote the unique (s,t)-path which contains the path constructed here and edges $\{s, g^{(i,2j)}\}$, $\{c_1^{(i,2j)}, c_2^{(i,2j)}\}$, $\{c_2^{(i,2j)}, c_1^{(i,2j+1)}\}$, $\{c_1^{(i,2j+1)}, c_2^{(i,2j+1)}\}$, $\{c_2^{(i,2j+1)}, h^{(i,2j)}\}$, and $\{h^{(i,2j)}, t\}$ by $P_l^{(i,2j)}$.
– Join $g^{(i,2j)}$ and $d_1^{(i,2j)}$ by a path of length 4 and add edges $\{d_2^{(i,2j)}, d_1^{(i,2j+1)}\}$, $\{d_2^{(i,2j+1)}, h^{(i,2j)}\}$. Denote the unique (s,t)-path which contains the path constructed here and edges $\{s, g^{(i,2j)}\}$, $\{d_1^{(i,2j)}, d_2^{(i,2j)}\}$, $\{d_2^{(i,2j)}, d_1^{(i,2j+1)}\}$, $\{d_1^{(i,2j+1)}, d_2^{(i,2j+1)}\}$, $\{d_2^{(i,2j+1)}, h^{(i,2j)}\}$, and $\{h^{(i,2j)}, t\}$ by $P_u^{(i,2j)}$.

We call the paths $P_l^{(i,j)}$ (resp. $P_u^{(i,j)}$) *lower* (resp. *upper*) *synchronizing* paths for F_i. Note that the total number of lower (upper) synchronizing paths for F_i is $r-1$. Also lower (resp. upper) paths for $F_i^{(1)}, \ldots, F_i^{(r)}$ are called *lower* (resp. *upper*) paths for F_i.

We need the following properties of the graph F_i constructed during the first 6 steps of the reduction.

Lemma 12. *The graph F_i has $2r-1$ edge-disjoint (s,t)-paths of length at most l, and any collection of such paths in F_i either contains r lower paths and $r-1$ upper synchronizing paths for F_i or contains r upper paths and $r-1$ lower synchronizing paths for F_i.*

Moreover, let F_i and a graph Q be induced subgraphs of a graph $Q' = F_i \cup Q$ such that

- $V(F_i) \cap V(Q) = \{s,t\} \cup \bigcup_{j=1}^{r} \{e_1^{(i,j)}, e_2^{(i,j)}, f_1^{(i,j)}, f_2^{(i,j)}\}$,
- *for every* $v \in \bigcup_{j=1}^{r} \{e_1^{(i,j)}, e_2^{(i,j)}, f_1^{(i,j)}, f_2^{(i,j)}\}$, $\mathrm{dist}_Q(v,t) > 5$.

Then any collection of $2r-1$ of edge-disjoint (s,t)-paths of length at most l in Q' which contain all edges of $\{\{s, x^{(i,1)}\}, \ldots, \{s, x^{(i,r)}\}\} \cup \{\{s, g^{(i,1)}\}, \ldots, \{s, g^{(i,r-1)}\}\}$ *either contains r lower paths and $r-1$ upper synchronizing paths for F_i or contains r upper paths and $r-1$ lower synchronizing paths for F_i.*

Proof. Observe at first that r lower paths and $r-1$ upper synchronizing paths for F_i (resp. r upper paths and $r-1$ lower synchronizing paths for F_i) are edge-disjoint (s,t)-paths of length at most l.

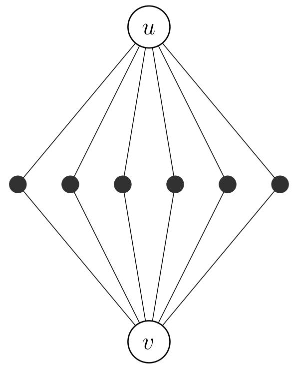


Fig. 4. The filter $L_6(u, v)$.

To prove the first claim of the lemma, note that for any collection $\mathcal{P}$ of $2r-1$ edge disjoint (s, t)-paths of length at most l in F_i, these paths go through all edges in the set $\{\{s, x^{(i,1)}\}, \dots, \{s, x^{(i,r)}\}\} \cup \{\{s, g^{(i,1)}\}, \dots, \{s, g^{(i,r-1)}\}\}$.

Consider (s, t)-paths in $\mathcal{P}$ which contain edges $\{s, g^{(i,1)}\}, \dots, \{s, g^{(i,r-1)}\}$. We show inductively that for each $j \in \{1, \dots, r-1\}$, the (s, t)-path of length at most l which goes through $\{s, g^{(i,j)}\}$ is either path $P_l^{(i,j)}$ or path $P_u^{(i,j)}$. If P is an (s, t)-path of length at most l which contains $\{s, g^{(i,1)}\}$ then either $P = P_l^{(i,1)}$ or $P = P_u^{(i,1)}$, since all other paths have bigger length. Let $j > 1$ and assume that paths in $\mathcal{P}$ which go through edges $\{s, g^{(i,1)}\}, \dots, \{s, g^{(i,j-1)}\}$ are paths from the set $\{P_l^{(i,1)}, \dots, P_l^{(i,j-1)}\} \cup \{P_u^{(i,1)}, \dots, P_u^{(i,j-1)}\}$. Since the first $j-1$ paths contain all edges $\{h^{(i,1)}, t\}, \dots, \{h^{(i,j-1)}, t\}$, the path $P \in \mathcal{P}$ which goes through $\{s, g^{(i,j)}\}$ cannot contain these edges and vertices $h^{(i,1)}, \dots, h^{(i,j-1)}$. Hence, P is either $P_l^{(i,j)}$ or $P = P_u^{(i,j)}$.

We proved that paths in $\mathcal{P}$ which include edges $\{s, g^{(i,1)}\}, \dots, \{s, g^{(i,r-1)}\}$ are synchronizing paths, and they contain all edges from the sets $\{\{s, g^{(i,1)}\}, \dots, \{s, g^{(i,r-1)}\}\}$ and $\{\{h^{(i,1)}, t\}, \dots, \{h^{(i,r-1)}, t\}\}$. Therefore for each $j \in \{1, \dots, r\}$, the (s, t)-path of length at most l which goes through $\{s, x^{(i,j)}\}$ cannot contain vertices $g^{(i,1)}, \dots, g^{(i,r-1)}$ and $h^{(i,1)}, \dots, h^{(i,r-1)}$. Hence, if P is a path in $\mathcal{P}$ which goes through $\{s, x^{(i,j)}\}$, then P is either the lower path for $F_i^{(j)}$ or the upper path for $F_i^{(j)}$. It remains to note that all $2r-1$ paths in $\mathcal{P}$ are edge-disjoint if and only if $\mathcal{P}$ either contains r lower paths for graphs $F_i^{(1)}, \dots, F_i^{(2)}$ and $r-1$ upper synchronizing paths $P_u^{(i,1)}, \dots, P_u^{(i,r-1)}$ or contains r upper paths for graphs $F_i^{(1)}, \dots, F_i^{(2)}$ and $r-1$ lower synchronizing paths $P_l^{(i,1)}, \dots, P_l^{(i,r-1)}$.

The second claim of the lemma is proved by the same arguments. It should only be noted that any (s, t)-path of length at most l in Q which goes through one of the edges of $\{\{s, x^{(i,1)}\}, \dots, \{s, x^{(i,r)}\}\} \cup \{\{s, g^{(i,1)}\}, \dots, \{s, g^{(i,r-1)}\}\}$ is a path in F_i. Suppose that P is an (s, t)-path of this type which contains edges of $E(Q') \setminus E(F_i)$. Since $V(F_i) \cap V(Q) = \{s, t\} \cup \bigcup_{j=1}^r \{e_1^{(i,j)}, e_2^{(i,j)}, f_1^{(i,j)}, f_2^{(i,j)}\}$, P contains as a segment an (s, u)-path P' in F_i for some $u \in \bigcup_{j=1}^r \{e_1^{(i,j)}, e_2^{(i,j)}, f_1^{(i,j)}, f_2^{(i,j)}\}$. By the construction of F_i, P' has length at least 6. The (u, t)-subpath of P should contain as a segment a (v, t)-path P'' for a vertex $v \in \bigcup_{j=1}^r \{e_1^{(i,j)}, e_2^{(i,j)}, f_1^{(i,j)}, f_2^{(i,j)}\}$ such that either P'' is a path in Q or a path in F_i. In the first case, P'' has length at least 6, by the conditions of the lemma, and therefore P has length at least 12. Assume that P'' is a path in F_i. Clearly, P'' has length at least 3. Since F_i is an induced subgraph of Q' and P contains edges of $E(Q') \setminus E(F_i)$, the (u, v)-subpath of P includes at least two edges. Hence P has length at least 11. $\square$

Let h be a positive integer. Denote by $L_h(u, v)$ the graph constructed from vertices u and v by joining them by h paths of length 2 as is shown in Fig. 4. Clearly, this graph has at most h edge-disjoint (u, v)-paths. Denote by $\mathcal{P}(L_h(u, v))$ the set of all these paths. We call $L_h(u, v)$ *h-filter*.

Let k be a positive integer. The next step of the reduction is the construction of the selection gadget H.

7. Construct graphs $F_1, \dots, F_n$ with common vertices s and t.
8. Add a vertex x, and construct a filter $L_{n-k}(s, x)$.
9. Add edges $\{x, f_1^{(1,1)}\}, \dots, \{x, f_1^{(n,1)}\}$.
10. Add a vertex y, and construct a filter $L_{n-k}(y, t)$.
11. Join y with vertices $f_2^{(1,1)}, \dots, f_2^{(n,1)}$ by paths $Q_1, \dots, Q_n$ of length 4.

We call a path $P \oplus \langle x, f_1^{(i,1)}, f_2^{(i,1)} \rangle \oplus Q_i \oplus P'$ where $P \in \mathcal{P}(L_{n-k}(s, x))$ and $P' \in \mathcal{P}(L_{n-k}(y, t))$ are the selection path for F_i. The next lemma concerns some properties of H.

Lemma 13. *The graph H has $n(2r-1) + n - k$ edge-disjoint (s, t)-paths of length at most l, and for any collection $\mathcal{P}$ of $n(2r-1) + n - k$ such paths, there is $I \subseteq \{1, \dots, n\}$, $|I| \le k$, such that*

- *all edges incident with s (resp. t) belong to different paths of $\mathcal{P}$,*
- *for $i \in I$, $\mathcal{P}$ contains r upper paths for F_i and $r-1$ lower synchronizing paths for F_i,*

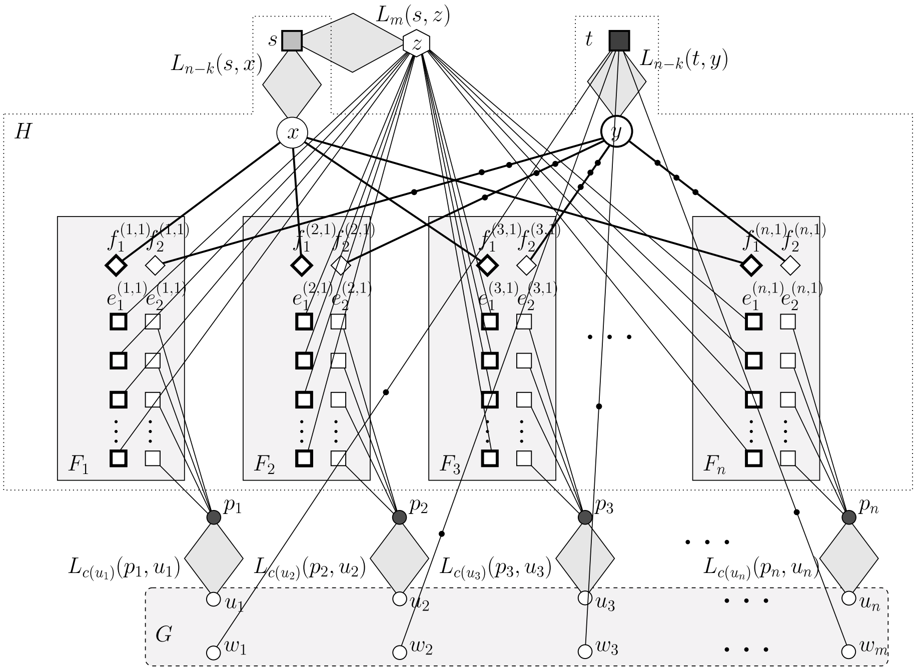


Fig. 5. The construction of the graph G'.

- *for $i \notin I$, $\mathcal{P}$ contains r lower paths for F_i and $r-1$ upper synchronizing paths for F_i,*
- *$\mathcal{P}$ contains $n-k$ selection paths for F_i such that $i \notin I$.*

Moreover, let H and a graph Q be induced subgraphs of a graph $Q' = H \cup Q$ such that

- *$V(H) \cap V(Q) = \{s,t\} \cup \bigcup_{i=1}^{n}(\{e_1^{(i,1)},\ldots,e_1^{(i,r)}\} \cup \{e_2^{(i,1)},\ldots,e_2^{(i,r)}\})$,*
- *for every $v \in \bigcup_{i=1}^{n}(\{e_1^{(i,1)},\ldots,e_1^{(i,r)}\} \cup \{e_2^{(i,1)},\ldots,e_2^{(i,r)}\})$, $\mathrm{dist}_Q(v,t) > 5$.*

Then any collection of $n(2r-1)+n-k$ edge-disjoint (s,t)-paths of length at most l in Q' which go through all edges incident with s in H satisfies the conditions listed above.

Proof. It is easy to check that for any $I \subseteq \{1,\ldots,n\}$, such that $|I| \le k$, any collection of paths described in the lemma is a collection of $n(2r-1)+n-k$ edge-disjoint (s,t)-paths of length at most l in H. Assume now that $\mathcal{P}$ is a collection of $n(2r-1)+n-k$ edge-disjoint (s,t)-paths of length at most l in H. Note that $\deg_H(s) = \deg_H(t) = n(2r-1)+n-k$ and therefore all edges incident with s and all edges incident with t belong to different paths from $\mathcal{P}$. By Lemma 12 there is $I \subseteq \{1,\ldots,n\}$ such that, for $i \in I$, $\mathcal{P}$ contains r upper paths and $r-1$ lower synchronizing paths for F_i, and, for $i \notin I$, $\mathcal{P}$ contains r lower paths and $r-1$ upper synchronizing paths for F_i. The remaining $n-k$ paths go through $L_{n-k}(s,x)$ and $L_{n-k}(y,t)$. It follows immediately that every such path is a selection path for some F_i. This path has to be edge-disjoint with upper paths for F_i. So, $i \notin I$ and $|I| \le k$.

The second claim of the lemma is proved by the same arguments. It should be noted only that (s,t)-paths in G which go through the filter $L_{n-k}(s,x)$ can only be selection paths. □

We complete our reduction as follows. Assume that $R = \{u_1,\ldots,u_n\}$ and $B = \{w_1,\ldots,w_m\}$.

12. Construct G and H.
13. Join each vertex w_i with t by a path W_i of length 2 for $i \in \{1,\ldots,m\}$.
14. For each $i \in \{1,\ldots,n\}$, vertex p_i is introduced and filters $L_{c(u_i)}(p_i,u_i)$ are constructed.
15. Add the edges $\{e_2^{(i,1)},p_i\},\ldots,\{e_2^{(i,r)},p_i\}$ for all $i \in \{1,\ldots,n\}$.
16. Construct the filter $L_m(s,z)$ and add the edges $\{z,e_1^{(i,1)}\},\ldots,\{z,e_1^{(i,r)}\}$ for all $i \in \{1,\ldots,n\}$.

Denote the obtained graph by G' (see Fig. 5). We call the path $P \oplus \langle z, e_1^{(i,h)}, e_2^{(i,h)}, p_i\rangle \oplus P' \oplus \langle u_i, w_j\rangle \oplus W_j$ the *i,j-dominating path* where $P \in \mathcal{P}(L_m(s,z))$, $P' \in \mathcal{P}(L_{c(u_i)}(p_i,u_i))$, and $h \in \{1,\ldots,r\}$.

The correctness of our reduction is based on the following lemma.

Lemma 14. *The red–blue capacitated graph (G,c) has a capacitated dominating set of size at most k if and only if G' has $k' = 2rn + m - k$ edge-disjoint (s,t)-paths of length at most l.*

Proof. Let $S \subseteq R$ be a capacitated dominating set in G of size at most k, and let f be a corresponding domination mapping. Set $I = \{i \in \{1, \ldots, n\}: u_i \in S\}$. We construct $n(2r-1)+n-k$ edge-disjoint (s,t)-paths as described in Lemma 13, i.e. we chose the collection of paths $\mathcal{P}$ such that for $i \in I$, $\mathcal{P}$ contains r upper paths and $r-1$ lower synchronizing paths for F_i, for $i \notin I$, $\mathcal{P}$ contains r lower paths and $r-1$ upper synchronizing paths for F_i, and $\mathcal{P}$ contains $n-k$ selection paths for F_i such that $i \notin I$. Now we add m dominating paths as follows. For each $j \in \{1, \ldots, m\}$, suppose that $f(w_j) = u_i$ and we add an i,j-dominating path. Such paths can be constructed since the number of paths which go through each pair of vertices u_i and p_i is at most $c(u_i) \leq r$.

Assume now that G' has a collection $\mathcal{P}$ of k' edge-disjoint (s,t)-paths of length at most l. Since $\deg_{G'}(s) = \deg_{G'}(t) = k'$, all edges incident with s (resp. t) belong to different paths. By Lemma 13, $n(2r-1)+n-k$ paths are paths in H which satisfy conditions given in the lemma, i.e. there is $I \subseteq \{1, \ldots, n\}$, $|I| \leq k$ such that for $i \in I$, $\mathcal{P}$ contains r upper paths for F_i and $r-1$ lower synchronizing paths for F_i, for $i \notin I$, $\mathcal{P}$ contains r lower paths for F_i and $r-1$ upper synchronizing paths for F_i and $\mathcal{P}$ contains $n-k$ selection paths for F_i such that $i \notin I$. We set $S = \{u_i \in R: i \in I\}$. It is easy to see that the remaining m paths can only be dominating ones. Observe also that since these paths have to be edge-disjoint with lower paths for F_i in $\mathcal{P}$, dominating paths can only go through vertices p_i for $i \in I$. Let $j \in \{1, \ldots, m\}$, and consider the dominating path $P \in \mathcal{P}$ which contains subpath W_j. The path P is an i,j-dominating path for some $i \in I$, and it goes through the filter $L_{c(u_i)}(p_i, u_i)$. Clearly, G' contains the edge $\{u_i, w_j\}$. We define a domination mapping for w_j by setting $f(w_j) = u_i$. It remains to note that we defined a domination mapping for all $w_j \in B$, and since at most $c(u_i)$ dominating paths can go through the filter $L_{c(u_i)}(p_i, u_i)$, at most $c(u_i)$ vertices are mapped to any vertex $u_i \in S$. □

To complete the proof of the theorem it remains to show that G' has bounded treewidth.

Lemma 15. *Let* $\mathbf{tw}(G) \leq t$. *Then* $\mathbf{tw}(G') \leq \max\{2(t+1)+4, 43\}$.

Proof. Suppose that $D = (\{X_i \mid i \in V(T)\}, T)$ is a tree decomposition of G of width at most t. We construct a tree decomposition of G' starting from it.

1. For each vertex $u_j \in V(G)$, add the vertex p_j to all bags of the tree decomposition which contain u_j. Note that the size of each bag after this operation is at most $2(t+1)$.
2. For each $j \in \{1, \ldots, n\}$, choose bag X_i which contains p_j and do the following:
   - Construct bags $Y_1^{(j)}, \ldots, Y_{r-1}^{(j)}$ such that $Y_h^{(j)}$ contains p_j, all vertices of $V(F_j^{(h)}) \cup V(F_j^{(h+1)}) \setminus \{s,t\}$, and all vertices of paths in $F_j - \{s,t\}$ with endpoints in $V(F_j^{(h)}) \cup V(F_j^{(h+1)}) \setminus \{s,t\}$. Note that $|Y_h^{(j)}| \leq 39$.
   - Add a path of length $r-1$ to T with one endpoint in i, and assign to other vertices of these paths bags $Y_1^{(j)}, \ldots, Y_{r-1}^{(j)}$.
3. Include vertices s, t, x, y, z in all bags.
4. Note that all remaining vertices which are not included to any bag belong to paths with endpoints in one bag and length of each path is at most 4. For each such a path we add to our tree a new leaf which contains all vertices of this path.

It can be easily verified by direct check that the obtained tree decomposition is indeed a tree decomposition of G', and it has width at most $\max\{2(t+1)+4, 43\}$. □

Part (b) of the theorem is proved by a similar construction. The only thing we have to do is to replace all undirected edges by direct ones using the rule that all lower, upper, synchronizing, selecting and dominating paths should be directed (s,t)-paths. □

5. Conclusions

A natural question about the parameterized complexity of the variants of the bounded length disjoint path and the bounded length cut problems parameterized by k and l is whether they admit polynomial kernels. In fact, using techniques from [35], we can prove that this is not the case for all the disjoint path variants unless the polynomial hierarchy collapses. We believe that the existence of polynomial kernels for the edge cut variants as well as the planar restrictions of the disjoint path variants is an interesting open problem.

Acknowledgements

We are grateful to the anonymous referees for their constructive suggestions and remarks. The first author was supported by EPSRC under project EP/G043434/1 and the second author was supported by the project "Kapodistrias" (AΠ 02839/28.07.2008) of the National and Kapodistrian University of Athens (project code: 70/4/8757).

References

[1] K. Menger, Über reguläre Baumkurven, Math. Ann. 96 (1927) 572–582.
[2] G.B. Dantzig, D.R. Fulkerson, On the max-flow min-cut theorem of networks, in: Linear Inequalities and Related Systems, in: Annals of Mathematics Studies, vol. 38, Princeton University Press, Princeton, NJ, 1956, pp. 215–221.

[3] L.R. Ford Jr., D.R. Fulkerson, Maximal flow through a network, Canad. J. Math. 8 (1956) 399–404.
[4] J. Adámek, V. Koubek, Remarks on flows in network with short paths, Comment. Math. Univ. Carolin. 12 (4) (1971) 661–667.
[5] L. Lovász, V. Neumann Lara, M. Plummer, Mengerian theorems for paths of bounded length, Period. Math. Hungar. 9 (1978) 269–276.
[6] G. Exoo, On line disjoint paths of bounded length, Discrete Math. 44 (1983) 317–318.
[7] L. Niepel, D. Šafaříková, On a generalization of Menger's theorem, Acta Math. Univ. Comenian. 42–43 (1983) 275–284.
[8] A. Itai, Y. Perl, Y. Shiloach, The complexity of finding maximum disjoint paths with length constraints, Networks 12 (1982) 277–286.
[9] G. Baier, T. Erlebach, A. Hall, E. Köhler, H. Schilling, M. Skutella, Length-bounded cuts and flows, in: Automata, Languages and Programming. Part I, in: Lecture Notes in Comput. Sci., vol. 4051, Springer, Berlin, 2006, pp. 679–690.
[10] G. Baier, T. Erlebach, A. Hall, E. Köhler, P. Kolman, O. Pangrác, H. Schilling, M. Skutella, Length-bounded cuts and flows, ACM Trans. Algorithms (in press).
[11] A. Bley, On the complexity of vertex-disjoint length-restricted path problems, Comput. Complexity 12 (2003) 131–149.
[12] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, M. Yannakakis, Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems, J. Comput. System Sci. 67 (2003) 473–496.
[13] L. Fleischer, M. Skutella, The quickest multicommodity flow problem, in: Integer Programming and Combinatorial Optimization, in: Lecture Notes in Comput. Sci., vol. 2337, Springer, Berlin, 2002, pp. 36–53.
[14] P. Kolman, C. Scheideler, Improved bounds for the unsplittable flow problem, in: Proceedings of the Symposium on Discrete Algorithms, ACM, 2002, pp. 184–193.
[15] P. Kolman, C. Scheideler, Improved bounds for the unsplittable flow problem, J. Algorithms 61 (1) (2006) 20–44.
[16] D. Ronen, Y. Perl, Heuristics for finding a maximum number of disjoint bounded paths, Networks 14 (1984) 531–544.
[17] D. Wagner, K. Weihe, A linear-time algorithm for edge-disjoint paths in planar graphs, Combinatorica 15 (1995) 135–150.
[18] D. Hsu, On container width and length in graphs, groups, and networks, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 77 (4) (1994) 668–680. Dedicated to Professor Paul Erdős on the occasion of his 80th birthday (Special Section on Discrete Mathematics and its Applications).
[19] R.M. Karp, On the computational complexity of combinatorial problems, Networks 5 (1) (1975) 45–68. (Proceedings of the Symposium on Large-Scale Networks, Evanston, IL, USA, 18–19 April 1974).
[20] N. Robertson, P.D. Seymour, Graph minors. XIII. The disjoint paths problem, J. Combin. Theory Ser. B 63 (1995) 65–110.
[21] S. Fortune, J. Hopcroft, J. Wyllie, The directed subgraph homeomorphism problem, Theoret. Comput. Sci. 10 (1980) 111–121.
[22] R.G. Downey, M.R. Fellows, Parameterized Complexity, in: Monographs in Computer Science, Springer-Verlag, New York, 1999.
[23] J. Flum, M. Grohe, Parameterized Complexity Theory, in: Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
[24] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, in: Oxford Lecture Series in Mathematics and its Applications, vol. 31, Oxford University Press, Oxford, 2006.
[25] C.-L. Li, T. McCormick, D. Simchi-Levi, The complexity of finding two disjoint paths with min–max objective function, Discrete Appl. Math. 26 (1990) 105–115.
[26] S. Tragoudas, Y.L. Varol, Computing disjoint paths with length constraints, in: Graph-Theoretic Concepts in Computer Science, Cadenabbia, 1996, in: Lecture Notes in Comput. Sci., vol. 1197, Springer, Berlin, 1997, pp. 375–389.
[27] N. Alon, R. Yuster, U. Zwick, Color-coding, J. Assoc. Comput. Mach. 42 (1995) 844–856.
[28] M. Fellows, D. Hermelin, F. Rosamond, S. Vialette, On the parameterized complexity of multiple-interval graph problems, Theoret. Comput. Sci. 410 (2009) 53–61.
[29] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms 12 (1991) 308–340.
[30] R.B. Borie, Generation of polynomial-time algorithms for some optimization problems on tree-decomposable graphs, Algorithmica 14 (1995) 123–137.
[31] J. Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, Discrete Appl. Math. 52 (1994) 233–252.
[32] M. Dom, D. Lokshtanov, S. Saurabh, Y. Villanger, Capacitated domination and covering: a parameterized perspective, in: IWPEC'08, in: LNCS, vol. 5018, Springer, 2008, pp. 78–90.
[33] H. Bodlaender, D. Lokshtanov, E. Penninkx, Planar capacitated dominating set is $W[1]$-hard, in: IWPEC'09, in: LNCS, vol. 5917, Springer, 2009, pp. 50–60.
[34] F.V. Fomin, P.A. Golovach, D. Lokshtanov, S. Saurabh, Algorithmic lower bounds for problems parameterized with clique-width, in: SODA'10, ACM-SIAM, 2010, pp. 493–502.
[35] H.L. Bodlaender, R.G. Downey, M.R. Fellows, D. Hermelin, On problems without polynomial kernels, J. Comput. System Sci. 75 (8) (2008) 423–434.
[36] P.A. Golovach, D.M. Thilikos, Paths of bounded length and their cuts: parameterized complexity and algorithms, in: IWPEC'09, in: LNCS, vol. 5917, Springer, 2009, pp. 210–221.