Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2017-07. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres. ISSN 1587–4451.

Blocking optimal structures

Kristóf Bérczi, Attila Bernáth, Tamás Király, and Gyula Pap

September 2017

Blocking optimal structures

Kristóf Bérczi*, Attila Bernáth*, Tamás Király*, and Gyula Pap**

Abstract

We consider weighted blocking problems (a.k.a. weighted transversal problems) of the following form. Given a finite set S, weights $w : S \to \mathbb{R}_+$, and a family $\mathcal{F} \subseteq 2^S$, find $\min\{w(H) : H \subseteq S,\ H$ intersects every member of $\mathcal{F}\}$. In our problems S is the set of edges of a (directed or undirected) graph and $\mathcal{F}$ is the family of optimal solutions of a combinatorial optimization problem with respect to a cost function $c : S \to \mathbb{R}_+$. Note that the cost function c that defines the family and the weight function w in the weighted transversal problem are not related.

In particular, we study the following four kinds of families: minimum cost k-spanning trees (unions of k edge-disjoint spanning trees), minimum cost s-rooted k-arborescences (unions of k arc-disjoint arborescences rooted at node s), and minimum cost (directed or undirected) k-braids between nodes s and t (unions of k edge-disjoint s-t paths). We focus on the special cases when either c or w is uniform. For the case $c \equiv 0$ (i.e. we want to block all combinatorial objects, not just the optimal ones), we show that most of the problems are NP-complete. In the other case, when $w \equiv 1$ (a minimum cardinality transversal problem for $\mathcal{F}$), most of our problems turn out to be polynomial-time solvable.

We also consider the problem of blocking k-edge-connectivity, which is related to both blocking k-spanning trees and blocking k-braids. We show that the undirected case can be solved in polynomial time using the ideas of Zenklusen's connectivity interdiction algorithm. In contrast, the directed version is shown to be NP-complete.

Keywords: minimum transversal, minimum weight transversal, k-spanning tree, k-arborescence, k-braid

*Department of Operations Research, Eötvös Loránd University, Budapest, Hungary. Email: {berkri,athos,tkiraly}@cs.elte.hu

**MTA-ELTE Egerváry Research Group, Eötvös Loránd University, Budapest, Hungary. Email: gyuszko@cs.elte.hu

1 Introduction

By **blocking problems** we mean the following type of problems. Given a finite set S and a family $\mathcal{F} \subseteq 2^S$, find $\min\{|H| : H \subseteq S,\ H$ intersects every member of $\mathcal{F}\}$. The family $\mathcal{F}$ consists of optimal solutions to some combinatorial optimization problem, for example minimum cost k-spanning trees of a graph (where S is the set of edges of a graph and a cost of each edge is given), or minimum cost k-arborescences of a digraph (where S is the set of arcs of a digraph and again we have a cost function on S). In the literature, these types of problems are also called **minimum transversal problems** for the family $\mathcal{F}$.

In a more general setting, we consider **weighted blocking problems** (or **minimum weight transversal problems**), that is, a weight function $w : S \to \mathbb{R}_+$ is also given and we want to find $\min\{w(H) : H$ intersects every member of $\mathcal{F}\}$. Note that this weight function is independent from the cost function that defines the family $\mathcal{F}$.

In particular, we will investigate the weighted blocking problem for four types of combinatorial structures: optimal k-spanning trees, optimal k-arborescences, and optimal undirected and directed k-braids. Let us define these objects.

Given an undirected graph $G = (V, E)$, a **k-spanning tree** is a subset B of edges that can be written as the union of k pairwise edge-disjoint spanning trees. It is known that k-spanning trees form the family of bases of a matroid, and that a minimum cost k-spanning tree can be found in polynomial time, if the cost $c(e)$ of each edge is given.

A **spanning arborescence** in a digraph $D = (V, A)$ is an arc set $F \subseteq A$ that is a spanning tree in the undirected sense and every node has in-degree at most one. Thus there is exactly one node, the **root node**, with in-degree zero. If the node set is clear from the context, spanning arborescences will be called **arborescences** for brevity. The arc-disjoint union of k spanning arborescences is called a **k-arborescence**. If every arborescence in the decomposition has the same root node s, then F is called an **s-rooted k-arborescence**. Given $D = (V, A)$, a positive integer k and a cost function $c : A \to \mathbb{R}_+$, a minimum cost k-arborescence or a minimum cost s-rooted k-arborescence can be found efficiently using the matroid intersection algorithm; see [17, Chapter 53.8] for a reference, where several related problems are considered. The existence of an s-rooted k-arborescence is characterized by Edmonds' disjoint arborescence theorem, while the existence of a k-arborescence is characterized by a theorem of Frank [7]. Frank also gave a linear programming description of the convex hull of k-arborescences, generalizing Edmonds' linear programming description of the convex hull of s-rooted k-arborescences.

Given an undirected graph $G = (V, E)$ and nodes s, t, an **undirected k-braid between s and t** is a subset of edges of G that can be decomposed into k pairwise edge-disjoint $s - t$ paths. Directed k-braids are defined analogously: in a digraph $D = (V, A)$, a **directed k-braid between nodes s and t** is a subset of arcs that can be decomposed into k pairwise arc-disjoint directed $s - t$ paths. We will use the term k-braid if we mean both the directed and undirected cases, or if the type of the graph is clear from context. Also, nodes s and t are omitted if they are clear from the context. It is known from network flow theory that we can find minimum cost (directed or undirected) k-braids if a non-negative cost $c(e)$ of every edge/arc is given.

1.1 Related work

Let us mention some known special cases of blocking problems. The **cuts** (or cocycles) of a matroid are the minimal transversals of the family of bases; in other words, a subset of the elements is a cut if it is an inclusionwise minimal subset that contains at least one element from each basis. The problem of finding minimum cuts in matroids has been studied in several different contexts (note the distinction between *minimal* and *minimum*: minimal is shorthand for *inclusionwise minimal*, while minimum means *minimum size*). Perhaps the best known special case is the minimum cut problem in graphs, which can be solved using network flows, and faster algorithms have also been developed (e.g. the Nagamochi-Ibaraki algorithm [15]). This corresponds to the minimum cardinality blocking problem for spanning trees; moreover, these methods also find the minimum weight cut, so they solve the minimum weight blocking problem for spanning trees, too.

For a matroid M and a positive integer k, let kM denote the matroid union of k identical copies of M. If M is a graphic matroid (or even a hypergraphic matroid, see [12]), then the minimum cut of kM can be found in polynomial time, even if k is part of the input. However, these methods do not extend to the minimum weight cut problem. Another notable open question is the complexity of finding a minimum cut in a rigidity matroid.

The minimum cut of a transversal matroid can also be found in polynomial time; however, the problem of finding a minimum *circuit* of a transversal matroid is NP-complete [14], which implies that the minimum cut problem is NP-complete for gammoids. Another line of research considers the minimum cut problem for binary matroids. Vardy [19] proved that the problem is NP-complete in general, but Geelen, Gerards, and Whittle [9] conjecture that the problem is in P for any minor-closed proper subclass of binary matroids. Partial results about this conjecture have been achieved by Geelen and Kapadia [10] and by Nägele, Sudakov, and Zenklusen [16].

The *minimum cost bases* (or *optimal bases* for brevity) of a matroid M form the bases of another matroid which can be obtained by taking the direct sum of certain minors of M. This means that we can find a minimum transversal of the family of optimal bases of M by solving minimum cut problems in some minors of M. In particular, if the minimum cut problem is solvable in polynomial time in a minor-closed class of matroids, then a minimum transversal of optimal bases can also be found in polynomial time. For example, the class of graphic matroids is minor-closed and the minimum cut problem can be solved efficiently, so we can also efficiently find a minimum transversal of the family of minimum spanning trees in a graph with edge costs.

The minimum transversal problem for arborescences can be formulated as the minimization of the sum of the in-degrees of two disjoint non-empty node sets of the digraph, which can be solved efficiently using network flows. The problem of finding a minimum transversal of the family of *minimum cost arborescences* is considerably more difficult. It can still be solved in polynomial time as shown in [4], but the solution requires more sophisticated tools than network flows. Finding a minimum transversal of the family of minimum cost k-arborescences is polynomial-time solvable for fixed k [2], but the problem is open if k is part of the input.

Let us finally mention that there is a lot of ongoing research on a related but slightly different class of problems called **interdiction problems**. In our terminology, an interdiction problem consists of finding $H \subseteq S$ with $w(H) \leq B$ for some fixed budget B, such that $\min\{c(X) : X \in \mathcal{F},\ X \cap H = \emptyset\}$ is as large as possible. For example, if $\mathcal{F}$ is the family of all spanning trees, then the aim is to find an edge set of weight at most B whose removal results in the highest increase in the minimum cost of a spanning tree. In contrast, in our problem the aim is to find the minimum weight edge set whose removal increases the minimum cost of a spanning tree by an arbitrarily small amount. Interdiction problems tend to be more difficult: spanning tree interdiction is NP-hard, and the best known approximation factor is 4 [13]. Network flow interdiction is related to blocking k-braids: the aim is to reduce the maximum $s-t$ flow in a network as much as possible, by removing edges of total weight at most B. This problem is also NP-hard, and Chestnut and Zenklusen [5] proved that an $n^{o(1)}$-approximation would imply an $n^{o(1)}$-approximation for the Densest k-Subgraph problem.

1.2 Our results

In this paper we consider the following problems.

Problem 1. *Given a graph $G = (V, E)$, cost function $c : E \to \mathbb{R}_+$, weight function $w : E \to \mathbb{R}_+$ and a positive integer k, find* $\min\{w(H) : H \subseteq E, H$ *intersects every c-optimal k-spanning tree*$\}$.

Problem 2. *Given a digraph $D = (V, A)$, cost function $c : A \to \mathbb{R}_+$, weight function $w : A \to \mathbb{R}_+$, node $s \in V$ and a positive integer k, find* $\min\{w(H) : H \subseteq A, H$ *intersects every c-optimal s-rooted k-arborescence*$\}$.

Problem 3. *Given a graph $G = (V, E)$, cost function $c : E \to \mathbb{R}_+$, weight function $w : E \to \mathbb{R}_+$, nodes $s, t \in V$ and a positive integer k, find* $\min\{w(H) : H \subseteq E, H$ *intersects every c-optimal k-braid from s to t*$\}$.

Problem 4. *Given a digraph $D = (V, A)$, cost function $c : A \to \mathbb{R}_+$, weight function $w : A \to \mathbb{R}_+$, nodes $s, t \in V$ and a positive integer k, find* $\min\{w(H) : H \subseteq A, H$ *intersects every c-optimal k-braid from s to t*$\}$.

We consider two types of restrictions on w and c. When $w \equiv 1$, i.e. w is uniform, our problems are **minimum cardinality transversal problems**, and they turn out to be polynomial-time solvable. The second type of restriction is $c \equiv 0$, that is, we want to **block all combinatorial objects**, not just the optimal ones. Note that $c \equiv 1$ could also be chosen for blocking all k-spanning trees or k-arborescences, but it is not suitable to describe all k-braids. With this restriction, most of our problems are NP-complete. We leave two questions open: we do not know the status of Problem 1 for $c \equiv 0$ (even for $k = 2$), and we do not know the status of Problem 2 if $w \equiv 1$. Our results are summarized in Table 1 below.

	Uniform weight ($w \equiv 1$)	**Uniform cost** ($c \equiv 0$)
Blocking optimal k-spanning trees (Problem 1)	**P** (Theorem 7)	Open (open even for $k = 2$)
Blocking optimal k-arborescences (Problem 2)	Open **P** for fixed k [2] **P** if $c \equiv 0, w \equiv 1$ [3]	**NPC** (Theorem 15) (**P** for fixed k [3])
Blocking optimal undirected k-braids (Problem 3)	**P** (Theorem 19)	**NPC** (Theorem 23) (**P** for fixed k, see Section 4)
Blocking optimal directed k-braids (Problem 4)	**P** (Theorem 21)	**NPC** (Theorem 23) (**P** for fixed k, see Section 4)

Table 1: Summary of results. **P** means that the problem is polynomial time solvable, **NPC** means that it is NP-complete.

1.3 Notation

Let us overview some of the notation and definitions used in the paper. Any notation not mentioned explicitly in this paper can be found in [8]. A **partition** $\mathcal{P}$ of a set V is a collection of pairwise disjoint non-empty subsets of V that together cover V. The partition is **trivial** if it consists of the single set V. We will use the notation $|\mathcal{P}|$ to denote the number of sets in the partition $\mathcal{P}$. A set family $\mathcal{L} \subseteq 2^V$ is said to be **laminar** if any two members of $\mathcal{L}$ are either disjoint or one is a subset of the other. For a function $x : A \to \mathbb{R}$ and subset $Z \subseteq A$, we use $x(Z) = \sum_{a \in Z} x_a$.

Given a (directed or undirected) graph $G = (V, E)$ and a subset $W \subseteq V$, let $G[W] = (W, \{uv \in E : u, v \in W\})$ be the restriction of G to W, and G/W be the graph obtained from G by contracting W into a single node (and deleting the loops that arise). If $B \subseteq E$ then we will also use $B[W]$ to mean the restriction of (V, B) to W and B/W to mean the contraction of W in (V, B). If $H \subseteq E$ then $G - H = (V, E - H)$ is the graph obtained from G by deleting the edges in H. Furthermore, if $\mathcal{L} \subseteq 2^V$ is a laminar family and $W \in \mathcal{L}$, then we denote by $\mathcal{L}/W$ the laminar family that is obtained from $\mathcal{L}$ by contracting W into a single node.

For a graph $G = (V, E)$ and some $Z \subseteq V$, $\delta_G(Z)$ denotes the set of edges in E with exactly one end-node in Z, and $d_G(Z) = |\delta_G(Z)|$ is the number of these edges. A graph G is said to be k-**edge-connected** if $d_G(Z) \geq k$ for every $\emptyset \neq Z \subsetneq V$.

For a digraph $D = (V, A)$ and some $Z \subseteq V$, $\delta^{in}_D(Z)$ and $\delta^{out}_D(Z)$ denote the set of arcs entering and leaving the set Z, respectively. We will also use the notation $\varrho_D(Z) = |\delta^{in}_D(Z)|$. A digraph D is said to be k-**arc-connected** if $\varrho_D(Z) \geq k$ for every $\emptyset \neq Z \subsetneq V$.

For two nodes $u, v \in V$, the notation uv will be used to denote an undirected edge between u and v, and also a directed arc from u to v. However, in order to make a clear distinction when necessary, we will sometimes write $\overrightarrow{uv}$ to denote a directed arc from u to v.

2 Blocking optimal k-spanning trees

For a graph $G = (V, E)$ and a partition $\mathcal{P}$ of the nodes of G, we denote by $e_G(\mathcal{P})$ the number of edges of G that go between two different classes of $\mathcal{P}$ (**cross-edges** in the partition $\mathcal{P}$). An undirected graph G is said to be (k, l)**-partition-connected** if $e_G(\mathcal{P}) \geq k(|\mathcal{P}| - 1) + l$ holds for any non-trivial partition $\mathcal{P}$ of the nodes of G.

A characterization for the existence of k edge-disjoint spanning trees in undirected graphs was given by Tutte.

Theorem 5 (Tutte, [18]). *A graph contains k edge-disjoint spanning trees if and only if it is $(k, 0)$-partition-connected.*

The (k, l)-partition-connectivity of a graph can be checked in polynomial time, as shown by the following result implicit in [8].

Theorem 6. *Given a graph $G = (V, E)$ and two positive integers k, l, we can decide in polynomial time if G is (k, l)-partition-connected or not. If it is not, then we can also find a partition $\mathcal{P}$ satisfying $e_G(\mathcal{P}) < k(|\mathcal{P}| - 1) + l$.*

Proof. If $l \geq k$, then G is (k, l)-partition-connected if and only if it is $(k + l)$-edge-connected [8, Proposition 1.2.11]. The $(k + l)$-edge-connectivity of G can be checked in polynomial time using network flows or the Nagamochi-Ibaraki minimum cut algorithm [15]. If the graph is not (k, l)-partition-connected then a minimum cut, considered as a partition with 2 classes, can serve as a witness.

If $l \leq k$, then the solution is described in [8], page 305. □

Using these results, we can solve Problem 1 in polynomial time in the special case when both c and w are uniform ($w \equiv 1$ and $c \equiv 0$): simply find (by logarithmic search) the smallest positive integer l such that G is not (k, l)-partition-connected along with a partition $\mathcal{P}$ satisfying $e_G(\mathcal{P}) < k(|\mathcal{P}| - 1) + l$. The optimal solution will be an arbitrary subset of cross-edges of $\mathcal{P}$ of size l (note that $e_G(\mathcal{P}) = k(|\mathcal{P}| - 1) + l - 1 \geq l$, as G is $(k, l - 1)$-partition-connected). This approach can be extended to deal with the case where c is not uniform.

Theorem 7. *Problem 1 is solvable in polynomial time if $w \equiv 1$.*

Proof. From the dual characterization of optimal k-spanning trees we get the following lemma.

Lemma 8. *Given a graph $G = (V, E)$, a positive integer k and a cost function $c : E \to \mathbb{R}_+$, we can find in polynomial time disjoint subsets $E_0, E_1 \subseteq E$ and a laminar family $\mathcal{L} \subseteq 2^V$ so that for any k-spanning tree $B \subseteq E$ the following statements are equivalent:*

1. *B is a c-optimal k-spanning tree,*

2. *$E_1 \subseteq B \subseteq E - E_0$ and $B[W]$ is a k-spanning tree of $G[W]$ for every $W \in \mathcal{L}$.* $\square$

We say that E_0 is the set of **forbidden edges**, while E_1 is the set of **mandatory edges**. Moreover, given a graph $G = (V, E)$ and a laminar family $\mathcal{L} \subseteq 2^V$, we say that a k-spanning tree $B \subseteq E$ is **$\mathcal{L}$-tight** if $B[W]$ is a k-spanning tree of $G[W]$ for every $W \in \mathcal{L}$. Note that $B \subseteq E$ is an $\mathcal{L}$-tight k-spanning tree if and only if it can be decomposed into k edge-disjoint $\mathcal{L}$-tight spanning trees. For later reference we state the following problem.

Problem 9 (Blocking $\mathcal{L}$-tight k-spanning trees)**.** *Given a graph $G = (V, E)$ and a laminar family $\mathcal{L} \subseteq 2^V$, find* $\min\{|H| : H$ *intersects every $\mathcal{L}$-tight k-spanning tree*$\}$.

Lemma 8 implies that the problem of blocking optimal k-spanning trees (Problem 1 for $w \equiv 1$) can be reduced to the problem of blocking $\mathcal{L}$-tight k-spanning trees. Indeed, if there are mandatory edges then we can block all optimal k-spanning trees by a single (mandatory) edge. Otherwise, we can just remove the forbidden edges, and the problem is to block $\mathcal{L}$-tight k-spanning trees in $G - E_0$. The rest of the proof is about the solution of Problem 9. We note that we can decide in polynomial time if an $\mathcal{L}$-tight k-spanning tree exists at all: this is a maximum cost k-spanning tree problem by setting the cost of an edge $uv \in E$ to be the number of sets in $\mathcal{L}$ that contain both endpoints of the edge – that is, $cost(uv) = |\{W \in \mathcal{L} : u, v \in W\}|$.

The following observation leads us to the solution of Problem 9.

Claim 10. *Given a graph $G = (V, E)$ and a laminar family $\mathcal{L} \subseteq 2^V$, let $W \in \mathcal{L}$ be an inclusionwise minimal member of $\mathcal{L}$. A subset $B \subseteq E$ is an $\mathcal{L}$-tight k-spanning tree if and only if $B[W]$ is a k-spanning tree in $G[W]$, and B/W is an $\mathcal{L}/W$-tight k-spanning tree in G/W.*

Proof: Clearly, if $B \subseteq E$ is an $\mathcal{L}$-tight k-spanning tree then $B[W]$ is a k-spanning tree in $G[W]$, and B/W is an $\mathcal{L}/W$-tight k-spanning tree in G/W.

On the other hand, assume that $B \subseteq E$ satisfies $B[W] = \dot{\bigcup}_{i=1}^{k} F_i^1$ and $B/W = \dot{\bigcup}_{i=1}^{k} F_i^2$ where each F_i^1 is a spanning tree of $G[W]$ and each F_i^2 is an $\mathcal{L}/W$-tight spanning tree in G/W. Then we can simply set $F_i = F_i^1 \cup F_i^2$ to obtain an $\mathcal{L}$-tight spanning tree in G for $i = 1, \ldots, k$. Thus $B = \dot{\bigcup}_{i=1}^{k} F_i$ is an $\mathcal{L}$-tight k-spanning tree in G, concluding the proof of the claim. $\bullet$

Using this claim, the solution to Problem 9 is the following. Pick an inclusionwise minimal member W of $\mathcal{L}$ and solve the problem of blocking all k-spanning trees in $G[W]$ (as described after Theorem 6) to get a candidate H_1. Then recursively solve the problem of blocking $\mathcal{L}/W$-tight k-spanning trees in G/W; let the minimum cardinality transversal be H_2. Finally, output the smaller of H_1 and H_2. $\square$

As an open problem we pose the following question: can we solve Problem 1 in polynomial time if c is uniform but w is not, that is, given a graph $G = (V, E)$, a positive integer k and $w : E \to \mathbb{R}_+$, can we determine $\min\{w(H) : H \subseteq E,\ G - H$ does not admit a k-spanning tree $\}$? We do not know how to solve this problem even for fixed k, e.g. $k = 2$.

2.1 A detour: blocking k-edge-connectivity

The weighted blocking problem of k-spanning trees for $k = 1$ (that is, Problem 1 for $c \equiv 0$ and $k = 1$) is equivalent to the minimum weight cut problem: given a graph $G = (V, E)$ and $w : E \to \mathbb{R}_+$, find $\min\{w(H) : H \subseteq E,\ G - H \text{ is not connected}\}$. This problem has another extension for larger k, namely the following **k-edge-connectivity blocking problem**.

Problem 11 (Blocking global k-edge-connectivity). *Given a graph $G = (V, E)$ and $w : E \to \mathbb{R}_+$, find $\min\{w(H) : H \subseteq E,\ G - H$ is not k-edge-connected$\}$.*

This fits in our framework by defining $\mathcal{F}$ as the family of all k-edge-connected spanning subgraphs of G. The problem is related to the uniform cost versions of both Problems 1 and 3. One important difference is that the structure of k-edge-connected subgraphs is much more complicated than the other structures considered in this paper – in particular, it is NP-hard to find a minimum cardinality 2-edge-connected spanning subgraph. In light of this, it is remarkable that while Problem 3 with $c \equiv 0$ is NP-complete (see Theorem 23), this problem is solvable in polynomial time.

Theorem 12. *Problem 11 is polynomial time solvable.*

Proof. The algorithm is analogous to the algorithm for connectivity interdiction developed by Zenklusen [20]. Let $e_1, e_2, \ldots, e_m$ be the enumeration of the edges ordered by increasing weight (where ties can be resolved arbitrarily), and let $E_i = \{e_1, \ldots, e_i\}$. For every $i \in [m]$, we solve the following problem: find $\min\{w(\delta_G(Z) \cap E_i) : \emptyset \neq Z \subsetneq V,\ |\delta_G(Z) \setminus E_i| \leq k - 1\}$. This is a bicriteria minimum cut problem, that can be solved in polynomial time using the method of Armon and Zwick [1]. Let $\ell \in [m]$ be the index for which the minimum is the smallest, and let Z be the core of the corresponding cut. We claim that $H = \delta_G(Z) \cap E_\ell$ is the optimal solution of the blocking problem. On one hand, removing H results in a graph that is not k-edge-connected because $|\delta_G(Z) \setminus H| \leq k - 1$. On the other hand, if H' is an optimal solution of the blocking problem, then there is a subset Z' such that H' contains the $d_G(Z') - k + 1$ edges with the smallest weight from $\delta_G(Z')$. We can thus assume that $H' = \delta_G(Z') \cap E_i$ for some i, and $|\delta_G(Z') \setminus E_i| \leq k - 1$. It follows that $w(H') = w(\delta_G(Z') \cap E_i) \geq w(\delta_G(Z) \cap E_\ell) = w(H)$, so H is also optimal. $\square$

The directed counterpart of Problem 11 turns out to be NP-complete.

Problem 13 (Blocking global k-arc-connectivity). *Given a digraph $D = (V, A)$, $w : E \to \mathbb{R}_+$ and a positive integer k, find $\min\{w(H) : H \subseteq A,\ D - H$ is not k-arc-connected$\}$.*

Theorem 14. *Problem 13 is NP-complete.*

The proof of this theorem will be given later, together with the proof of Theorem 23. Note that a simple brute force algorithm can solve Problem 13 in polynomial time if k is not part of the input.

3 Blocking optimal k-arborescences

Problem 2 for $k = 1$ was solved in [4]. For $w \equiv 1$, an algorithm solving Problem 2 was given in [2] that has polynomial running time if k is fixed. If both w and c are uniform, then the problem is polynomially solvable even if k is part of the input [3]. Furthermore, it was observed in [3] that for uniform c and fixed k the problem is solvable in polynomial time (with a simple brute force technique). In this light it is somewhat surprising that Problem 2 is NP-complete for $c \equiv 0$, if k is part of the input.

Theorem 15. *Problem 2 is NP-complete in the special case $c \equiv 0$.*

The proof of this theorem will be given later, together with the proof of Theorem 23.

4 Blocking optimal k-braids

Given a digraph $D = (V, A)$, $s, t \in V$, $k \in \mathbb{Z}_+$ and a cost function $c : A \to \mathbb{R}_+$, the following theorem characterizes optimal directed k-braids.

Theorem 16 (Ford and Fulkerson [6])**.** *The minimum cost of a directed k-braid from s to t is equal to*

$$\max\left\{k\pi(t) + \sum[c_\pi(uv) : uv \in A,\ c_\pi(uv) < 0] : \pi \in \mathbb{R}_+^V,\ \pi(s) = 0\right\}, \tag{1}$$

where $c_\pi(uv) = c(uv) - \pi(v) + \pi(u)$ for every arc $uv \in A$.

Corollary 17. *We can find in polynomial time a partition $A = A_- \cup A_0 \cup A_+$ so that a k-braid $F \subseteq A$ is optimal if and only if $A_- \subseteq F \subseteq A - A_+$.*

Proof. Choose an optimal solution π^* of (1) and let $A_- = \{uv \in A : c_{\pi^*}(uv) < 0\}$, $A_0 = \{uv \in A : c_{\pi^*}(uv) = 0\}$, and $A_+ = \{uv \in A : c_{\pi^*}(uv) > 0\}$. Note that π^* and thus A_-, A_0, A_+ can be found in polynomial time (see eg. [8, Theorem 3.6.1]). The complementary slackness conditions imply that a k-braid $F \subseteq A$ is optimal if and only if $A_- \subseteq F \subseteq A - A_+$. □

4.1 Relationship between the directed and the undirected problems

It is an easy observation that inclusionwise minimal directed k-braids do not contain directed cycles. In order to deal with the undirected case, we need a similar statement on inclusionwise minimal transversals of optimal directed k-braids.

Lemma 18. *Given a digraph $D = (V, A)$, $s, t \in V$, $k \in \mathbb{Z}_+$ and a cost function $c : A \to \mathbb{R}_+$, if $H \subseteq A$ is an inclusionwise minimal arc set that intersects every optimal k-braid, then H does not contain a directed cycle.*

Proof. Suppose, for contradiction, that there exists a directed cycle $C = \{f_1, \dots, f_\ell\}$ in H. By the minimality of H, there exists a c-optimal k-braid B_i with $B_i \cap H = \{f_i\}$ for $i = 1, \dots, \ell$. Let $B = \cup_{i=1}^{\ell} B_i - C$ and define a capacity function $g : B \to \mathbb{Z}_+$ by setting $g(f) = |\{i : f \in B_i\}|$. Note that $g(f) \le \ell$ for every arc $f \in B$.

We claim that we can pack ℓ k-braids $B'_1, \dots, B'_\ell$ in B under the capacities. Indeed, it is known (see e.g. [17, (13.12)]) that the convex hull P of incidence vectors of those subsets of B that contain k arc-disjoint $s-t$ paths is determined by

$$\begin{aligned} 0 \le x(a) \le 1 \qquad & \text{for each } a \in B, \\ x(Q) \ge k \qquad & \text{for each } s-t \text{ cut } Q. \end{aligned}$$

By a result of L. E. Trotter [17, Theorem 13.8], the polytope P described by these inequalities has the so-called integer decomposition property, meaning that for each $\ell \in \mathbb{Z}_+$, any integer vector $x \in \ell \cdot P$ is the sum of ℓ integer vectors in P. Clearly, $g \in \ell \cdot P$, hence the existence of $B'_1, \dots, B'_\ell$ follows.

By the optimality of the B_is, $\sum_{i=1}^{\ell} c(B_i) = c(C) + \sum_{i=1}^{\ell} c(B'_i) \ge \sum_{i=1}^{\ell} c(B'_i) \ge \sum_{i=1}^{\ell} c(B_i)$. Thus equality must hold throughout and so B'_i is c-optimal for $i = 1, \dots, \ell$, contradicting the assumption that H is a blocking arc-set. □

Now we turn to the problem of blocking undirected optimal k-braids.

Theorem 19. *Problem 3 can be reduced to Problem 4 in polynomial time.*

Proof. Consider an instance of Problem 3 given by a graph $G = (V, E)$ and cost and weight functions $c, w : E \to \mathbb{R}_+$. We define a digraph $G^\circ = (V, E^\circ)$ and cost and weight functions $c^\circ, w^\circ : E^\circ \to \mathbb{R}_+$ as follows. For each edge $e = uv$ of G, add a pair of symmetric arcs $e' = \overrightarrow{uv}$ and $e'' = \overrightarrow{vu}$ to E° with cost $c^\circ(e') = c^\circ(e'') = c(e)$ and weight $w^\circ(e') = w^\circ(e'') = w(e)$. Let $\tau = \min\{w(H) : H \subseteq E, H$ intersects every c-optimal k-braid from s to t in $G\}$ be the optimum in G, while $\tau^\circ = \min\{w^\circ(H) : H \subseteq E^\circ, H$ intersects every c°-optimal k-braid from s to t in $G^\circ\}$ be the optimum in G°. The proof is completed by the following claim.

Claim 20. $\tau = \tau^\circ$. *Moreover, an optimal blocking set in G° can be transformed to an optimal blocking set in G.*

Proof: Let $H^\circ \subseteq E^\circ$ be an optimal blocking set in G°. Let $H = \{uv \in E : uv$ or $vu \in H^\circ\}$. Clearly, H covers every c-optimal k-braid in G and $w(H) \le w^\circ(H^\circ)$, hence $\tau \le \tau^\circ$ (in fact, $w(H) = w^\circ(H^\circ)$ by Lemma 18, if H° is also an inclusionwise minimal solution).

To see the other direction, take an optimal blocking set $H \subseteq E$ in G. Let $H^\circ = \{\overrightarrow{uv}, \overrightarrow{vu} \in E^\circ : uv \in H\}$. Now H° covers every c°-optimal k-braid in G°. Note that $w^\circ(H^\circ) = 2w(H) = 2\tau$ as H° contains both e' and e'' for each $e \in H$. However, by Lemma 18, H° contains a minimal blocking set that contains at most one of e' and e'' for each $e \in H$. This shows $\tau \ge \tau^\circ$, thus concluding the proof of the claim. • □

4.2 Uniform weight

Based on Theorem 16, first we show how the minimum cardinality blocking of c-optimal directed k-braids can be solved in polynomial time. The undirected case then follows from the directed one.

Theorem 21. *Problem 4 is solvable in polynomial time in the special case $w \equiv 1$.*

Proof. Find the 3-partition $A_- \cup A_0 \cup A_+$ of A as in Corollary 17. By the corollary, the following algorithm solves the problem.

Case 1: $A_- \neq \emptyset$ In this case the optimal k-braids can be blocked by a single arc from A_-.

Case 2: $A_- = \emptyset$ In this case an optimal solution consists of all-but-$(k-1)$ arcs from a minimum $s-t$ cut in $D_0 = (V, A_0)$. That is, the minimum number of arcs blocking all c-optimal directed k-braids is

$$\min\{\varrho_{A_0}(Z) - (k-1) : t \in Z \subseteq V - s\}.$$

This concludes the proof of the theorem. □

Theorems 19 and 21 together imply the following corollary.

Corollary 22. *Problem 3 can be solved in polynomial time if $w \equiv 1$.*

4.3 NP-completeness of the weighted versions

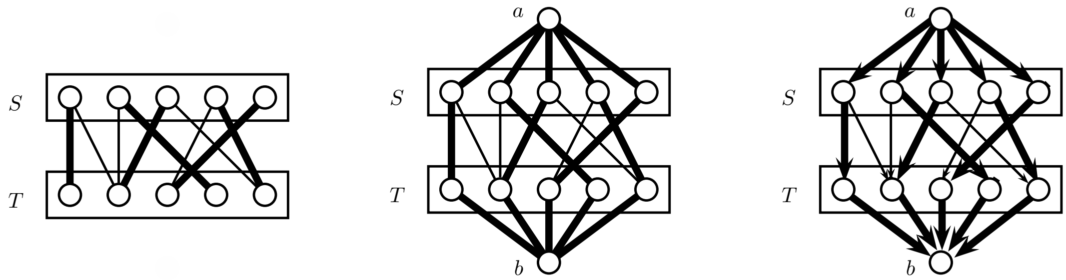
In contrast to the polynomial-time solvability of the minimum cardinality blocking problem of minimum cost k-braids, the weighted blocking problems for k-braids are NP-complete even if $c \equiv 0$.

Theorem 23. *Problems 3 and 4 are both NP-complete in the special case $c \equiv 0$.*

Proof of Theorems 14, 15 and 23: Clearly, the decision versions of the problems are in NP, therefore we will only concentrate on proving their completeness.

Given a bipartite graph $G_0 = (S, T, E_0)$ with $|S| = |T| = k$, consider the following constructions (see Figure 1).

1. Let $a, b \notin S \cup T$ be new nodes and let $V = S \cup T \cup \{a, b\}$. Let $G = (V, E)$ where $E = E_0 \cup \{as : s \in S\} \cup \{tb : t \in T\}$.
2. Let $D_1 = (V, A_1)$ be the digraph obtained from G (defined above) by orienting each edge "from a to b" – that is, $A_1 = \{\overrightarrow{as} : s \in S\} \cup \{\overrightarrow{st} : st \in E_0\} \cup \{\overrightarrow{tb} : t \in T\}$.
3. Let $D_2 = (V, A_2)$ be obtained from D_1 by adding k parallel arcs from b to every $v \in S \cup T$.



(a) Original graph (b) Graph G (c) Digraph D_1

Figure 1: Constructions for G and D_1

4. Let $D_3 = (V, A_3)$ be obtained from D_2 by adding k parallel arcs from every $v \in S \cup T$ to a.

Claim 24. *The following statements are equivalent.*

(i) G_0 *admits a perfect matching.*

(ii) There is an undirected k*-braid from* a *to* b *in* G*.*

(iii) There is a directed k*-braid from* a *to* b *in* D_1*.*

(iv) There is an a*-rooted* k*-arborescence in* D_2*.*

(v) D_3 *is* k*-arc-connected.*

PROOF: It is quite straightforward how (i) implies all of the other items in the list above (see the thick edges in Figure 1 for an illustration). On the other hand, if G_0 does not have a perfect matching, then by Hall's theorem there is a subset $X \subseteq S$ with $|\Gamma_{G_0}(X)| < |X|$, and then the set $a + X + \Gamma_{G_0}(X)$ defines a cut that shows that neither of (ii)-(v) can hold. •

The proof of the theorem can be finished as follows. We will reduce the following problem.

Problem 25 (Blocking Bipartite Matchings)**.** *Given a bipartite graph* $G_0 = (S, T, E_0)$*, find* $\min\{|H| : H \subseteq E_0,\ G_0 - H$ *does not have a perfect matching*$\}$.

It is known (see e.g. [11]) that Problem 25 is NP-complete. Given an instance $G_0 = (S, T, E_0)$ of this problem, we construct the graph G and digraphs D_1, D_2 and D_3 as above, and set the weights as follows. The (directed or undirected) edges $st \in E_0$ have weight 1, while any other edge has a large weight M (for example $M = |E_0| + 1$ suffices). Thus we have defined an instance of Problem 3 with input G, an instance of Problem 4 with input D_1 and an instance of Problem 2 with input D_2: in all three cases we have also defined weights for edges/arcs, and the cost function is defined to be zero in all three cases. We have also defined an instance of Problem 13 with digraph D_3 and the given weight function. Note that $k = |S| = |T|$ in all the defined problems. Then the problem of Blocking Bipartite Matchings in G_0 has a solution of size m if and only if either of the defined weighted blocking problems has a solution of total weight at most m. $\square$

Note that both Problems 3 and 4 can be solved in polynomial time if $c \equiv 0$ and k is fixed, using a brute-force search technique, similar to the one used in [3] for solving Problem 2 for fixed k and $c \equiv 0$.

Acknowledgement

The research was supported by the MTA-ELTE Egerváry Research Group and the Hungarian National Research, Development and Innovation Office – NKFIH, grants K109240 and K120254.

References

[1] A. Armon and U. Zwick. Multicriteria global minimum cuts. *Algorithmica*, 46(1):15–26, 2006.

[2] A. Bernáth and T. Király. Blocking optimal k-arborescences. Technical Report TR-2015-09, Egerváry Research Group, Budapest, 2015. `www.cs.elte.hu/egres`.

[3] A. Bernáth and G. Pap. Blocking unions of arborescences. *Discrete Optimization*, 22, Part B:277 – 290, 2016.

[4] A. Bernáth and G. Pap. Blocking optimal arborescences. *Mathematical Programming*, 161(1):583–601, 2017.

[5] S. R. Chestnut and R. Zenklusen. Hardness and approximation for network flow interdiction. *Networks*, 69(4):378–387, 2017.

[6] D. R. Ford and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, NJ, USA, 1962.

[7] A. Frank. On disjoint trees and arborescences. In *Algebraic Methods in Graph Theory, Colloquia Mathematica Soc. J. Bolyai*, volume 25, pages 159–169, 1978.

[8] A. Frank. *Connections in Combinatorial Optimization.* Oxford Lecture Series in Mathematics and Its Applications. OUP Oxford, 2011.

[9] J. Geelen, B. Gerards, and G. Whittle. The highly connected matroids in minor-closed classes. *Annals of Combinatorics*, pages 1–17, 2013.

[10] J. Geelen and R. Kapadia. Computing girth and cogirth in perturbed graphic matroids. *Combinatorica*, Jan 2017.

[11] G. Joret and A. Vetta. Reducing the rank of a matroid. *Discrete Mathematics & Theoretical Computer Science*, Vol. 17 no.2, Jan. 2015.

[12] T. Király. Computing the minimum cut in hypergraphic matroids. Technical Report QP-2009-05, Egerváry Research Group, Budapest, 2009. `www.cs.elte.hu/egres`.

[13] A. Linhares and C. Swamy. Improved Algorithms for MST and Metric-TSP Interdiction. *ArXiv e-prints*, May 2017.

[14] S. T. McCormick. A combinatorial approach to some sparse matrix problems. Technical report, DTIC Document, 1983.

[15] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5(1):54–66, 1992.

[16] M. Nägele, B. Sudakov, and R. Zenklusen. Submodular Minimization Under Congruency Constraints. *ArXiv e-prints*, July 2017.

[17] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Verlag, 2003.

[18] W. T. Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, 1(1):221–230, 1961.

[19] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997.

[20] R. Zenklusen. Connectivity interdiction. *Operations Research Letters*, 42(67):450 – 454, 2014.