Discrete Optimization 22 (2016) 277–290

Contents lists available at ScienceDirect

Discrete Optimization

www.elsevier.com/locate/disopt

Blocking unions of arborescences

Attila Bernáth[1], Gyula Pap*

MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, Budapest, H-1117, Hungary

ARTICLE INFO

Article history:
Received 30 July 2015
Received in revised form 11 August 2016
Accepted 16 August 2016
Available online 22 September 2016

Keywords:
Arborescences
Polynomial algorithm
Minimum transversal

ABSTRACT

Given a digraph $D = (V, A)$ and a positive integer k, a subset $B \subseteq A$ is called a k-arborescence, if it is the disjoint union of k spanning arborescences. When also arc-costs $c : A \to \mathbb{R}$ are given, minimizing the cost of a k-arborescence is well-known to be tractable. In this paper we take on the following problem: what is the minimum cardinality of a set of arcs the removal of which destroys every minimum c-cost k-arborescence. Actually, the more general weighted problem is also considered, that is, arc weights $w : A \to \mathbb{R}_+$ (unrelated to c) are also given, and the goal is to find a minimum weight set of arcs the removal of which destroys every minimum c-cost k-arborescence. An equivalent version of this problem is where the roots of the arborescences are fixed in advance. In an earlier paper (Bernáth and Pap, 2013) we solved this problem for $k = 1$. This work reports on other partial results on the problem. We solve the case when both c and w are uniform—that is, find a minimum size set of arcs that covers all k-arborescences. Our algorithm runs in polynomial time for this problem. The solution uses a result of Bárász et al. (2005) saying that the family of so-called in-solid sets (sets with the property that every proper subset has a larger in-degree) satisfies the Helly-property, and thus can be (efficiently) represented as a subtree hypergraph. We also give an algorithm for the case when only c is uniform but w is not. This algorithm is only polynomial if k is not part of the input.

1. Introduction

Given a hypergraph on ground set S with hyperedge set $\mathcal{E}$, a **transversal** of $\mathcal{E}$ is an inclusionwise minimal subset of S that intersects every member of $\mathcal{E}$. The set of transversals of $\mathcal{E}$ is also called the **blocker** of $\mathcal{E}$. Our problems can be viewed as finding a minimum transversal of a certain hypergraph, or more generally, finding a minimum weight transversal, if weights of the elements of S are also given. The problem of finding a minimum (weight) transversal of $\mathcal{E}$ will also be called **the blocking problem for** $\mathcal{E}$. Notice the difference

* Corresponding author.
E-mail addresses: bernath@cs.elte.hu (A. Bernáth), gyuszko@cs.elte.hu (G. Pap).

[1] Part of the research was done while the author was at Warsaw University, Institute of Informatics, ul. Banacha 2, 02-097 Warsaw, Poland.

http://dx.doi.org/10.1016/j.disopt.2016.08.003
1572-5286/© 2016 Published by Elsevier B.V.

between the terms *minimal* and *minimum*: minimal means inclusionwise minimal, while minimum means minimum size.

Let $D = (V, A)$ be a digraph with vertex set V and arc set A. A **spanning arborescence** is a subset $B \subseteq A$ 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. Equivalently, a spanning arborescence is a subset $B \subseteq A$ with the property that there is a root node $r \in V$ such that $\varrho_B(r) = 0$, and $\varrho_B(v) = 1$ for $v \in V - r$, and B contains no cycle. (Here $\varrho_B(v)$ denotes the number of arcs in B entering node v.) We will also call a spanning arborescence an **arborescence** for short, when the set of nodes is obvious from the context. If $r \in V$ is the root of the spanning arborescence B then B is said to be an r-**rooted arborescence**.

Given also a positive integer k, a subset $B \subseteq A$ is called a k-**arborescence**, if it is the arc-disjoint union of k spanning arborescences. In the special case when every arborescence has the same root r, we call B an r-**rooted** k-**arborescence**.

Given $D = (V, A)$, k and a cost function $c : A \to \mathbb{R}$, it is well known how to find a **minimum cost** r-**rooted** k-**arborescence** in polynomial time, and to find a **minimum cost** k-**arborescence** just as well. See [1, Chapter 53.8] for a reference, where several related problems are considered. The existence of an r-rooted k-arborescence is characterized by Edmonds' Disjoint Arborescence Theorem (Theorem 2), while the existence of a k-arborescence is characterized by a theorem of Frank [2] (see Theorem 3). Frank also gave a linear programming description of the convex hull of k-arborescence, generalizing Edmonds' linear programming description of the convex hull of r-rooted k-arborescences. The problem of finding a minimum cost k-arborescence may also be solved with the use of these results, either via a reduction to minimum cost r-rooted k-arborescences, or by applying a weighted matroid intersection algorithm directly.

In this paper we consider the following blocking problems, and actually, we will prove that these problems are polynomial time equivalent.

Problem 1 (***Blocking Optimal*** k-***Arborescences***)**.** Given a digraph $D = (V, A)$, a positive integer k, a cost function $c : A \to \mathbb{R}$ and a nonnegative weight function $w : A \to \mathbb{R}_+$, find a subset H of the arc set such that H intersects every minimum c-cost k-arborescence, and $w(H)$ is minimum.

Here the expression "intersects" simply means that the two have non-empty intersection. We remark that it is quite easy to see that Problem 1 is polynomially equivalent with the version where the root is also given in advance, that is, the problem of blocking optimal r-rooted k-arborescences.

Problem 2 (***Blocking Optimal*** r-***Rooted*** k-***Arborescences***)**.** Given a digraph $D = (V, A)$, a node $r \in V$, a positive integer k, a cost function $c : A \to \mathbb{R}$ and a nonnegative weight function $w : A \to \mathbb{R}_+$, find a subset H of the arc set such that H intersects every minimum c-cost r-rooted k-arborescence, and $w(H)$ is minimum.

In Section 4 we will show that the two problems are polynomial time equivalent. However, if c is constant then Problems 1 and 2 are *not* easily seen to be equivalent: at least we did not find such reductions (find more details in Section 5). Thus it is important to distinguish between these two versions, if c is constant.

Problems 1 and 2 can be viewed as hypergraph blocking problems. The ground set of the hypergraph is the arc set A of a digraph for both problems, and the family of hyperedges is the family of minimum c-cost k-arborescences for Problem 1, and the family of minimum c-cost r-rooted k-arborescences for Problem 2.

In our previous paper [3] we have solved Problems 1 and 2 in the special case when $k = 1$. Our conjecture is that both problems are also polynomial time solvable when k is not fixed. The main result of this paper is that both problems are polynomial time solvable when k is part of the input, and both c and w are constant—that is, when H is required to intersect *every* k-arborescence and we want to minimize $|H|$. Our