Constructing Minimum Changeover Cost Arborescenses in Bounded Treewidth Graphs *

Didem Gözüpek[1], Hadas Shachnai[2], Mordechai Shalom[3,4], and Shmuel Zaks[2]

[1]Department of Computer Engineering, Gebze Techical University, Kocaeli, Turkey
didem.gozupek@gtu.edu.tr
[2]Department of Computer Science, Technion, Haifa 32000, Israel. [hadas,zaks]@cs.technion.ac.il
[3]TelHai College, Upper Galilee, 12210, Israel cmshalom@telhai.ac.il
[4]Department of Industrial Engineering, Bogaziçi University, Istanbul, Turkey

Given an edge-colored graph, a vertex on a path experiences a reload cost if it lies between two consecutive edges of different colors. The value of the reload cost depends only on the colors of the traversed edges. Reload cost has important applications in dynamic networks, such as transportation networks and dynamic spectrum access networks. In the *minimum changeover cost arborescence (*MinCCA*)* problem, we seek a spanning tree of an edge-colored graph, in which the total sum of crossing all internal vertices, starting from a given root, is minimized. In general, MinCCA is known to be hard to approximate within factor $n^{1-\epsilon}$, for any $\epsilon > 0$, on a graph of n vertices.

We first show that MinCCA can be optimally solved in polynomial-time on cactus graphs. Our main result is an optimal polynomial-time algorithm for graphs of bounded *treewidth*, thus establishing first evidence to the solvability of our problem on a fundamental subclass of graphs. Our results imply that MinCCA is *fixed parameter tractable* when parameterized by treewidth and the maximum degree of the input graph.

Keywords: Reload cost, changeover cost, cactus graphs, bounded treewidth graphs, network design

1 Introduction

Background: Reload cost in an edge-colored graph refers to the cost incurred when crossing a vertex along a path where the incident edges have distinct colors. The value of the reload cost depends on the colors of the crossed edges. Two common models relate to this concept: the *reload cost* model, and the *changeover cost* model. In the former, the cost is proportional

*This work is supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant no. 113E567, and by TUBITAK 2221 Programme.

to the amount of commodity flowing through the edges, whereas in the latter, the cost is independent of the amount of commodity. The reload cost concept has important applications in transportation networks, dynamic spectrum access networks, telecommunication and energy distribution networks. Different modes of transportation can be represented by different edge colors. Loading and unloading of cargo at the transfer points has a significant cost, which can be represented using reload costs.

Related Work: The notion of reload cost was introduced in [7]. This and other papers (e.g. [1, 3, 5]) study various network problems that aim to minimize a global cost related to the reload cost. The paper [4] studies the problem of finding a spanning tree that minimizes the sum of reload costs over the paths between *all* pairs of vertices, and proves the NP-hardness of the problem. The paper [2] studies a closely related yet different problem, called *minimum changeover cost arborescence (* MinCCA*)*. The goal is to find a spanning tree that minimizes the sum of the reload costs of traversing all internal vertices, where the sum is independent of the amount of commodity traversing the edges. This problem is the focus of our study. The authors consider directed graphs and show that the problem is hard to approximate within a factor of $n^{\frac{1}{3}-\epsilon}$, for any $\epsilon > 0$, when there are 3 colors. In [6], we derived several inapproximability results, as well as polynomial-time algorithms for some special cases of MinCCA. In particular, we showed that MinCCA is polynomial-time solvable for the special case of cactus graphs where the number of cycles each vertex can belong to is bounded by some constant.

Our Contribution: In this paper, we study the minimum changeover cost arborescence problem on *undirected* graphs. In solving MinCCA, our algorithms for cactus graphs, and for graphs of bounded treewidth, make non-trivial use of dynamic programming. To the best of our knowledge, the complexity of MinCCA in graphs of bounded treewidth is studied here for the first time. Moreover, except for the results in [6], we are not aware of any previous studies of MinCCA on special graph classes. Given the hardness results for general graphs, an important contribution of this paper is in establishing first evidence to the polynomial solvability of MinCCA on a fundamental subclass of graphs. Our results imply that MinCCA is *fixed parameter tractable* when parameterized by treewidth and the maximum degree of the input graph. The question whether MinCCA parameterized by the treewidth alone is in FPT remains open.

2 Preliminaries

Graphs, digraphs, trees, forests: Given an undirected graph $G = (V(G), E(G))$, $d_G(v)$ is the degree of v in G. The minimum and maximum degrees of G are denoted as $\delta(G)$ and $\Delta(G)$ respectively. We denote by $N_G(U)$ (resp. $N_G[U]$) the open (resp. closed) neighborhood of U in G. $N_G(U)$ is the set of vertices of $V(G) \setminus U$ that are adjacent to a vertex of U, and $N_G[U] \stackrel{def}{=} N_G(U) \cup U$. When there is no ambiguity about the graph G we omit it from the subscripts. For a subset of vertices $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G induced by U. The degree of a vertex in a digraph is its degree in the underlying graph. A vertex is a *source* of a digraph G if there is no arc entering it. We denote the set of sources of a digraph G by $SRC(G)$. The *inbound induced subgraph* $G[U, in]$ is the subgraph of G that consists of all arcs incoming to a vertex of U and all the vertices that are endpoints of these arcs.

We denote by $parent_T(v)$ the *parent* of v in the rooted tree T. A *rooted forest* is the disjoint union of rooted trees. Clearly, the number of sources of a rooted forest is equal to the number of the trees. We use the tree notation also for forests, whenever appropriate.

The changeover cost: We follow the notation and terminology of [7] where the concept of reload cost was defined. We consider edge colored graphs G, where the colors are taken from a finite set X and $\chi : E(G) \mapsto X$ is the *coloring function*. The costs are given by a non-negative function $c : X^2 \mapsto \mathbb{N}_0$ satisfying $\forall x_1, x_2 \in X, c(x_1, x_2) = c(x_2, x_1)$, and $\forall x \in X, c(x, x) = 0$.

The cost of traversing two incident edges e_1, e_2 is $c(e_1, e_2) \stackrel{def}{=} c(\chi(e_1), \chi(e_2))$. Given a tree T rooted at r, for every outgoing edge of r we define $prev(e) = e$, and for every other edge $prev(e)$ is the edge preceding e on the path from r to e. The changeover cost of T with respect to r is $c(T, r) \stackrel{def}{=} \sum_{e \in E(T)} c(prev(e), e)$.

Problem Statement: The MinCCA problem aims to find a spanning tree rooted at r with minimum changeover cost [2]. Formally,

MinCCA (G, X, χ, r, c)
Input: A graph $G = (V, E)$ with an edge coloring function $\chi : E \mapsto X$, a vertex $r \in V$ and a reload cost function $c : X^2 \mapsto \mathbb{N}_0$.
Output: A spanning tree T of G.
Objective: Minimize $c(T, r)$.

Parameterized complexity: In parameterized complexity theory, the complexity of an algorithm is expressed as a function of both the size n of the input, and a parameter k depending on the input. A problem is called *fixed parameter tractable* (FPT), if it can be solved in time $f(k) \cdot p(n)$, where f is a function depending solely on k and p is a polynomial in n.

3 Bounded Treewidth Graphs

In this section we present a polynomial-time algorithm for MinCCA for any graph whose treewidth is bounded by some constant. Our algorithms make non-trivial use of dynamic programming. This algorithm is an extension of a simpler algorithm for cactus graphs which is omitted from this presentation.

We start by introducing notations that we use in this section. Let G be the input graph, and let $\mathcal{T}$ be a small tree decomposition of G of smallest width. For simplicity we direct $\mathcal{T}$ so that it is rooted at some bag B that contains r. All trees and forests considered in this section are rooted. The ancestor-descendant relation among the vertices of a rooted forest is clearly a partial order. We refer to a forest also as the partial order it induces. Two forests on the same set of vertices are equal if and only if the partial orders they induce are equal. Given a forest F and $U \subseteq V(F)$, $F\big|_U$ is a forest with vertex set U, s.t. (u, u') is an arc of $F\big|_U$ if and only if u is the closest ancestor of u' in F such that $u \in U$. In other words, the partial order $F\big|_U$ is the partial order F restricted to U.

Proposition 1. *$SRC(F) \cap U \subseteq SRC(F\big|_U)$. Moreover, if $SRC(F) \subseteq U$ then $SRC(F) = SRC(F\big|_U)$.*

For a bag $B \in V(\mathcal{T})$ we denote by $\mathcal{T}_B$ the subtree of $\mathcal{T}$ rooted at B, and define $B^* \stackrel{def}{=} \cup\mathcal{T}_B$ as all the vertices in the bags of $\mathcal{T}_B$. For a subset U of vertices of G, $\mathcal{F}(U)$ is the set of all non-trivial forests F with the following properties: The source of F is not in U but is a neighbor of some vertex in U; in addition, F spans all vertices in U and naturally its source vertices (which are not in U). More formally,

$$\mathcal{F}(U) \stackrel{def}{=} \{F \text{ is a forest} : U \subseteq V(F) \subseteq N_G[U], SRC(F) = V(F) \setminus U, \delta(F) > 0\}.$$

We note that every forest of $\mathcal{F}(U)$ can be obtained by choosing a parent for every $u \in U$ from its neighbors. Therefore, $|\mathcal{F}(U)| \leq \prod_{u \in U} d_G(u) \leq \Delta(G)^{|U|}$.

For a forest $F \in \mathcal{F}(B)$ we define $\mathcal{F}(F)$ as follows:

$$\mathcal{F}(F) \stackrel{def}{=} \{f \text{ is a forest} : V(f) = SRC(F), SRC(f) = SRC(F) \setminus B^*\}.$$

We note that every forest of $\mathcal{F}(F)$ can be obtained by choosing a parent for every $v \in SRC(F) \cap B^*$ from $SRC(F)$. Therefore, $|\mathcal{F}(F)| \leq (|SRC(F)| - 1)^{|SRC(F)|}$.

Let $F \in \mathcal{F}(B)$, and $f \in \mathcal{F}(F)$. We say that a forest F' is compatible with F and f, and denote as $F' \sim (F, f)$ if $F \cup F'$ is a forest and $(F \cup F')\big|_{V(f)}$ is a subgraph of f. We denote as $c_U(F)$ the cost of those edges that join two vertices of U in forest F. The following lemma describes how a tree decomposition divides a spanning tree into forests.

Lemma 2. *Let B be a bag of $\mathcal{T}$ with children $B_1, \ldots B_k$ in $\mathcal{T}$. $F^* \in \mathcal{F}(B^*)$ if and only if there exist unique forests $F, f, F_1^*, \ldots, F_k^*$ such that*
i) $F^* = F \cup \cup_{i \in [k]} F_i^*$, ii) $F \in \mathcal{F}(B)$, iii) $\forall i \in [k], F_i^* \in \mathcal{F}(B_i^*)$, iv) $f \in \mathcal{F}(F)$, v) $\forall i \in [k], F_i^* \sim (F, f)$, and
v) $c(F^*) = c(F) + \sum_{i=1}^{k} \left(c(F_i^*) - c_Y(F_i^*)\right)$ where $Y = B \cap (\cup_{i \in [k]} B_i)$.

For a forest $F \in \mathcal{F}(B)$, and $f \in \mathcal{F}(F)$, let $opt(B, F, f)$ be the minimum cost of a forest $F^* \in \mathcal{F}(B^*)$ that is compatible with (F, f). The following lemma implies a dynamic programming algorithm.

Lemma 3. *Let B be a bag of $\mathcal{T}$ with children $B_1, \ldots B_k$, $F \in \mathcal{F}(B)$, and $f \in \mathcal{F}(F)$. Then,*

$$opt(B, F, f) = c(F) + \sum_{i=1}^{k} \min_{F_i \in \mathcal{F}(B_i)} \left(\min_{f_i \in \mathcal{F}(F_i), (F_i \cup f_i) \sim (F, f)} opt(B_i, F_i, f_i) - c_Y(F_i) \right)$$

where Y is defined as in Lemma 2.

Theorem 4. *There exists a dynamic programming algorithm solving* MinCCA *optimally for bounded treewidth graphs and its running time is $O(\Delta(G)^{2(tw(G)+1)})$.*

References

[1] E. Amaldi, G. Galbiati, and F. Maffioli. On minimum reload cost paths, tours, and flows. *Networks*, 57(3):254–260, 2011.

[2] G. Galbiati, S. Gualandi, and F. Maffioli. On minimum changeover cost arborescences. In *Experimental Algorithms - 10th International Symposium (SEA), Kolimpari, Chania, Crete, Greece*, pages 112–123, May 2011.

[3] G. Galbiati, S. Gualandi, and F. Maffioli. On minimum reload cost cycle cover. *Discrete Applied Mathematics*, 164:112–120, 2014.

[4] I. Gamvros, L. Gouveia, and S. Raghavan. Reload cost trees and network design. *Networks*, 59(4):365–379, 2012.

[5] L. Gourvès, A. Lyra, C. Martinhon, and J. Monnot. The minimum reload s-t path, trail and walk problems. *Discrete Applied Mathematics*, 158(13):1404–1417, July 2010.

[6] D. Gözüpek, M. Shalom, A. Voloshin, and S. Zaks. On the complexity of constructing minimum changeover cost arborescences. *Theoretical Computer Science*, 540–541:40–52, 2014.

[7] H. Wirth and J. Steffan. Reload cost problems: minimum diameter spanning tree. *Discrete Applied Mathematics*, 113(1):73–85, 2001.