Parameterized Complexity of the MINCCA Problem on Graphs of Bounded Decomposability

Didem Gözüpek[1], Sibel Özkan[2], Christophe Paul[3], Ignasi Sau[3](✉), and Mordechai Shalom[4,5]

[1] Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey
didem.gozupek@gtu.edu.tr
[2] Department of Mathematics, Gebze Technical University, Kocaeli, Turkey
s.ozkan@gtu.edu.tr
[3] CNRS, LIRMM, Université de Montpellier, Montpellier, France
{paul,sau}@lirmm.fr
[4] TelHai College, Upper Galilee 12210, Israel
cmshalom@telhai.ac.il
[5] Department of Industrial Engineering, Boğaziçi University, Istanbul, Turkey

Abstract. In an edge-colored graph, the cost incurred at a vertex on a path when two incident edges with different colors are traversed is called reload or changeover cost. The *Minimum Changeover Cost Arborescence* (MinCCA) problem consists in finding an arborescence with a given root vertex such that the total changeover cost of the internal vertices is minimized. It has been recently proved by Gözüpek *et al.* [14] that the MinCCA problem is FPT when parameterized by the treewidth and the maximum degree of the input graph. In this article we present the following results for MinCCA:

- the problem is W[1]-hard parameterized by the treedepth of the input graph, even on graphs of average degree at most 8. In particular, it is W[1]-hard parameterized by the treewidth of the input graph, which answers the main open problem of [14];
- it is W[1]-hard on multigraphs parameterized by the tree-cutwidth of the input multigraph;
- it is FPT parameterized by the star tree-cutwidth of the input graph, which is a slightly restricted version of tree-cutwidth. This result strictly generalizes the FPT result given in [14];
- it remains NP-hard on planar graphs even when restricted to instances with at most 6 colors and 0/1 symmetric costs, or when restricted to instances with at most 8 colors, maximum degree bounded by 4, and 0/1 symmetric costs.

Keywords: Minimum Changeover Cost Arborescence · Parameterized complexity · FPT algorithm · Treewidth · Dynamic programming · Planar graph

This work is supported by the bilateral research program of CNRS and TUBITAK under grant no.114E731.

M. Shalom—The work of this author is supported in part by the TUBITAK 2221 Programme.

P. Heggernes (Ed.): WG 2016, LNCS 9941, pp. 195–206, 2016.
DOI: 10.1007/978-3-662-53536-3_17

1 Introduction

The cost that occurs at a vertex when two incident edges with different colors are crossed over is referred to as *reload cost* or *changeover cost* in the literature. This cost depends on the colors of the traversed edges. Although the reload cost concept has important applications in numerous areas such as transportation networks, energy distribution networks, and cognitive radio networks, it has received little attention in the literature. In particular, reload/changeover cost problems have been investigated very little from the perspective of parameterized complexity; the only previous work we are aware of is the one in [14].

In heterogeneous networks in telecommunications, transiting from a technology such as 3G (third generation) to another technology such as wireless local area network (WLAN) has an overhead in terms of delay, power consumption etc., depending on the particular setting. This cost has gained increasing importance due to the recently popular concept of vertical handover [6], which is a technique that allows a mobile user to stay connected to the Internet (without a connection loss) by switching to a different wireless network when necessary. Likewise, switching between different service providers even if they have the same technology has a non-negligible cost. Recently, cognitive radio networks (CRN) have gained increasing attention in the communication networks research community. Unlike other wireless technologies, CRNs are envisioned to operate in a wide range of frequencies. Therefore, switching from one frequency band to another frequency band in a CRN has a significant cost in terms of delay and power consumption [2,13]. This concept has applications in other areas as well. For instance, the cost of transferring cargo from one mode of transportation to another has a significant cost that outweighs even the cost of transporting the cargo from one place to another using a single mode of transportation [19]. In energy distribution networks, transferring energy from one type of carrier to another has an important cost corresponding to reload costs [8].

The reload cost concept was introduced in [19], where the considered problem is to find a spanning tree having minimum diameter with respect to reload cost. In particular, they proved that the problem cannot be approximated within a factor better than 3 even on graphs with maximum degree 5, in addition to providing a polynomial-time algorithm for graphs with maximum degree 3. The work in [8] extended these inapproximability results by proving that the problem is inapproximable within a factor better than 2 even on graphs with maximum degree 4. When reload costs satisfy the triangle inequality, they showed that the problem is inapproximable within any factor better than 5/3.

The work in [10] focused on the minimum reload cost cycle cover problem, which is to find a set of vertex-disjoint cycles spanning all vertices with minimum total reload cost. They showed an inapproximability result for the case when there are 2 colors, the reload costs are symmetric and satisfy the triangle inequality. They also presented some integer programming formulations and computational results.

The authors in [12] study the problems of finding a path, trail or walk connecting two given vertices with minimum total reload cost. They present several polynomial and NP-hard cases for (a)symmetric reload costs and reload costs with(out) triangle inequality. Furthermore, they show that the problem is polynomial for walks, as previously mentioned by [19], and re-proved later for directed graphs by [1].

The work in [9] introduced the *Minimum Changeover Cost Arborescence* (MinCCA) problem. Given a root vertex, MinCCA problem is to find an arborescence with minimum total changeover cost starting from the root vertex. They proved that even on graphs with bounded degree and reload costs adhering to the triangle inequality, MinCCA on directed graphs is inapproximable within $\beta \log\log(n)$ for $\beta > 0$ when there are two colors, and within $n^{1/3-\epsilon}$ for any $\epsilon > 0$ when there are three colors. The work in [15] investigated several special cases of the problem such as bounded cost values, bounded degree, and bounded number of colors. In addition, [15] presented inapproximability results as well as a polynomial-time algorithm and an approximation algorithm for the considered special cases.

In this paper, we study the MinCCA problem from the perspective of parameterized complexity; see [3,5,7,17]. Unlike the classical complexity theory, parameterized complexity theory takes into account not only the total input size n, but also other aspects of the problem encoded in a parameter k. It mainly aims to find an exact resolution of NP-complete problems. A problem is called *fixed parameter tractable* (FPT) if it can be solved in time $f(k) \cdot p(n)$, where $f(k)$ is a function depending solely on k and $p(n)$ is a polynomial in n. An algorithm constituting such a solution is called an FPT algorithm for the problem. Analogously to NP-completeness in classical complexity, the theory of W[1]-hardness can be used to show that a problem is unlikely to be FPT, i.e., for every algorithm the parameter has to appear in the exponent of n. The parameterized complexity of reload cost problems is largely unexplored in the literature. To the best of our knowledge, [14] is the only work that focuses on this issue by studying the MinCCA problem on bounded treewidth graphs. In particular, [14] showed that the MinCCA problem is in XP when parameterized by the treewidth of the input graph and it is FPT when parameterized by the treewidth *and* the maximum degree of the input graph. In this paper, we prove that the MinCCA problem is W[1]-hard parameterized by the treedepth of the input graph, even on graphs of average degree at most 8. In particular, it is W[1]-hard parameterized by the treewidth of the input graph, which answers the main open issue pointed out by [14]. Furthermore, we prove that it is W[1]-hard on multigraphs parameterized by the tree-cutwidth of the input multigraph. On the positive side, we present an FPT algorithm parameterized by the star tree-cutwidth of the input graph, which is a slightly restricted version of tree-cutwidth that we introduce here. This algorithm strictly generalizes the FPT algorithm given in [14]. We also prove that the problem is NP-hard on planar graphs, which are also graphs of bounded decomposability, even when restricted to instances with at most 6 colors and 0/1 symmetric costs. In addition, we prove that it remains NP-hard on planar graphs even when restricted to instances with at most 8 colors, maximum degree bounded by 4, and 0/1 symmetric costs.

The rest of this paper is organized as follows. In Sect. 2 we introduce some basic definitions and preliminaries as well as a formal definition of the MinCCA problem. We present our hardness results in Sect. 3. Finally, Sect. 4 concludes the paper. Due to space limitations, the proofs of the results marked with '[$\star$]', our algorithmic results with respect to star tree-cutwidth, as well as several figures, can be found in the full version of the article, which is permanently available at [arXiv:1605.00532].

2 Preliminaries

We say that two partial functions f and f' *agree* if they have the same value everywhere they are both defined, and we denote it by $f \sim f'$. For a set A and an element x, we use $A + x$ (resp., $A - x$) as a shorthand for $A \cup \{x\}$ (resp., $A \setminus \{x\}$). We denote by $[i, k]$ the set of all integers between i and k inclusive, and $[k] = [1, k]$.

Graphs, Digraphs, Trees, and Forests. Given an undirected (multi)graph $G = (V(G), E(G))$ and a subset $U \subseteq V(G)$ of the vertices of G, $\delta_G(U) := \{uu' \in E(G) \mid u \in U, u' \notin U\}$ is the *cut* of G determined by U, i.e., the set of edges of G that have exactly one end in U. In particular, $\delta_G(v)$ denotes the set of edges incident to v in G, and $d_G(v) := |\delta_G(v)|$ is the *degree* of v in G. The *minimum and maximum degrees* of G are defined as $\delta(G) := \min\{d_G(v) \mid v \in V(G)\}$ and $\Delta(G) := \max\{d_G(v) \mid v \in V(G)\}$ 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 $N_G[U] := N_G(U) \cup U$. 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.

A digraph T is a *rooted tree* or *arborescence* if its underlying graph is a tree and it contains a *root* vertex denoted by $\mathsf{root}(T)$ with a directed path from every other vertex to it. Every other vertex $v \neq \mathsf{root}(T)$ has a parent in T, and v is a *child* of its parent.

A rooted forest is the disjoint union of rooted trees, that is, each connected component of it has a root, which will be called a *sink* of the forest.

Tree Decompositions, Treewidth, and Treedepth. A *tree decomposition* of a graph $G = (V(G), E(G))$ is a tree $\mathcal{T}$, where $V(\mathcal{T}) = \{B_1, B_2, \ldots\}$ is a set of subsets (called *bags*) of $V(G)$ such that the following three conditions are met:

1. $\bigcup V(\mathcal{T}) = V(G)$.
2. For every edge $uv \in E(G)$, $u, v \in B_i$ for some bag $B_i \in V(\mathcal{T})$.
3. For every $B_i, B_j, B_k \in V(\mathcal{T})$ such that B_k is on the path $P_{\mathcal{T}}(B_i, B_j)$, $B_i \cap B_j \subseteq B_k$.

The *width* $\omega(\mathcal{T})$ of a tree decomposition $\mathcal{T}$ is defined as the size of its largest bag minus 1, i.e., $\omega(\mathcal{T}) = \max\{|B| \mid B \in V(\mathcal{T})\} - 1$. The *treewidth* of a graph G, denoted as $\mathbf{tw}(G)$, is defined as the minimum width among all tree decompositions of G. When the treewidth of the input graph is bounded, many efficient algorithms are known for problems that are in general NP-hard. In fact, most problems are known to be FPT when parameterized by the treewidth of the input graph. Hence, what we prove in this paper, i.e., the MINCCA problem is W[1]-hard when parameterized by treewidth, is an interesting result.

The *treedepth* $\mathbf{td}(G)$ of a graph G is the smallest natural number k such that each vertex of G can be labeled with an element from $\{1, \ldots, k\}$ so that every path in G joining two vertices with the same label contains a vertex having a larger label. Intuitively, where the treewidth parameter measures how far a graph is from being a tree, treedepth measures how far a graph is from being a star. The treewidth of a graph is at most one less than its treedepth; therefore, a W[1]-hardness result for treedepth implies a W[1]-hardness for treewidth.

Tree-Cutwidth. We now explain the concept of tree-cutwidth and follow the notation in [11]. A *tree-cut decomposition* of a graph G is a pair $(T, \mathcal{X})$ where T is a rooted tree and $\mathcal{X}$ is a near-partition of $V(G)$ (that is, empty sets are allowed) where each set X_t of the partition is associated with a node t of T. That is, $\mathcal{X} = \{X_t \subseteq V(G) : t \in V(T)\}$. The set X_t is termed the *bag* associated with the *node* t. For a node t of T we denote by Y_t the union of all the bags associated with t and its descendants, and $G_t = G[Y_t]$. $\mathsf{cut}(t) = \delta(Y_t)$ is the set of all edges with exactly one endpoint in Y_t.

The *adhesion* $\mathsf{adh}(t)$ of t is $|\mathsf{cut}(t)|$. The *torso* of t is the graph H_t obtained from G as follows. Let $t_1, \ldots, t_\ell$ be the children of t, $Y_i = Y_{t_i}$ for $i \in [\ell]$ and $Y_0 = V(G) \setminus (X_t \cup_{i=1}^{\ell} Y_i)$. We first contract each set Y_i to a single vertex y_i, by possibly creating parallel edges. We then remove every vertex y_i of degree 1 (with its incident edge), and finally *suppress* every vertex y_i of degree 2 having 2 neighbors, by connecting its two neighbors with an edge and removing y_i. The *torso size* $\mathsf{tor}(t)$ of t is the number of vertices in H_t. The *width* of a tree-cut decomposition $(T, \mathcal{X})$ of G is $\max_{t \in V(T)}\{\mathsf{adh}(t), \mathsf{tor}(t)\}$. The *tree-cutwidth* of G, or $\mathbf{tcw}(G)$ in short, is the minimum width of $(T, \mathcal{X})$ over all tree-cut decompositions $(T, \mathcal{X})$ of G.

Figure 1 shows the relationship between the graph parameters that we consider in this article. As depicted in Fig. 1, tree-cutwidth provides an intermediate measurement which allows either to push the boundary of fixed parameter tractability or strengthen W[1]-hardness result (cf. [11,16,20]). Furthermore, Fig. 1 also shows that treedepth and tree-cutwidth are unrelated.

Reload and Changeover Costs. We follow the notation and terminology of [19] 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 $\chi : E(G) \rightarrow X$ is the *coloring function*. Given a coloring function χ, we denote by E_x^{χ}, or simply by E_x the set of edges of E colored x, and $G_x = (V(G), E(G)_x)$ is the subgraph of G having the same vertex set as G, but only the edges colored x. The costs are given by a non-negative function $cc : X^2 \rightarrow \mathbb{N}_0$ satisfying

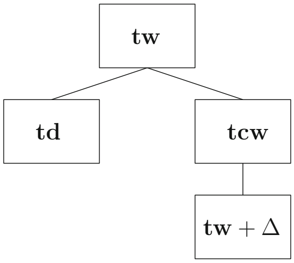

Fig. 1. Relationships between several graph parameters. A being a child of B means that every graph class with bounded A has also bounded B [11]

1. $cc(x_1, x_2) = cc(x_2, x_1)$ for every $x_1, x_2 \in X$.
2. $cc(x, x) = 0$ for every $x \in X$.

The cost of traversing two incident edges e_1, e_2 is $cc(e_1, e_2) := cc(\chi(e_1), \chi(e_2))$. The changeover cost of a path $P = (e_1 - e_2 - \ldots - e_\ell)$ of length ℓ is $cc(P) := \sum_{i=2}^{\ell} cc(e_{i-1}, e_i)$. Note that $cc(P) = 0$ whenever $\ell \leq 1$.

We extend this definition to trees as follows: Given a directed tree T rooted at r, (resp., an undirected tree T and a vertex $r \in V(T)$), for every outgoing edge e of r (resp., incident to 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 $cc(T, r) := \sum_{e \in E(T)} cc(prev(e), e)$. When there is no ambiguity about the vertex r, we denote $cc(T, r)$ by $cc(T)$.

Statement of the Problem. The MinCCA problem aims to find a spanning tree rooted at r with minimum changeover cost [9]. Formally,

> MinCCA
> **Input:** A graph $G = (V, E)$ with an edge coloring function $\chi : E \to X$, a vertex $r \in V$ and a changeover cost function $cc : X^2 \to \mathbb{N}_0$.
> **Output:** A spanning tree T of G minimizing $cc(T, r)$.

3 Hardness Results

In this section we prove several hardness results for the MinCCA problem. Our main result is in Subsect. 3.1, where we prove that the problem is W[1]-hard parameterized by the treedepth of the input graph. We also prove that the problem is W[1]-hard on *multigraphs* parameterized by the tree-cutwidth of the input graph. Both results hold even if the input graph has bounded average degree. Finally, in Subsect. 3.2 we prove that the problem remains NP-hard on planar graphs.

3.1 W[1]-hardness with Parameters Treedepth and Tree-Cutwidth

We need to define the following parameterized problem.

> MULTICOLORED k-CLIQUE
> **Input:** A graph G, a coloring function $c : V(G) \to \{1, \dots, k\}$, and a positive integer k.
> **Parameter:** k.
> **Question:** Does G contain a clique on k vertices with one vertex from each color class?

MULTICOLORED k-CLIQUE is known to be W[1]-hard on general graphs, even in the special case where all color classes have the same number of vertices [18], and therefore we may make this assumption as well.

Theorem 1. *The* MINCCA *problem is* W[1]*-hard parameterized by the treedepth of the input graph, even on graphs with average degree at most 8.*

Proof. We reduce from MULTICOLORED k-CLIQUE, where we may assume that k is odd. Indeed, given an instance (G, c, k) of MULTICOLORED k-CLIQUE, we can trivially reduce the problem to itself as follows. If k is odd, we do nothing. Otherwise, we output $(G', c', k+1)$, where G' is obtained from G by adding a universal vertex v, and $c' : V(G') \to \{1, \dots, k+1\}$ is such that its restriction to G equals c, and $c(v) = k + 1$.

Given an instance (G, c, k) of MULTICOLORED k-CLIQUE with k odd, we proceed to construct an instance (H, X, χ, r, cc) of MINCCA. Let $V(G) = V_1 \uplus V_2 \uplus \cdots \uplus V_k$, where the vertices of V_i are colored i for $1 \leq i \leq k$. Let W be an arbitrary Eulerian circuit of the complete graph K_k, which exists since k is odd. If $V(K_k) = \{v_1, \dots, v_k\}$, we can clearly assume without loss of generality[1] that W starts by visiting, in this order, vertices $v_1, v_2, \dots, v_k, v_1$, and that the last edge of W is $\{v_3, v_1\}$. For every edge $\{v_i, v_j\}$ of W, we add to H a vertex $s_{i,j}$. These vertices are called the *selector vertices* of H. For every two consecutive edges $\{v_i, v_j\}, \{v_j, v_\ell\}$ of W, we add to H a vertex $v_j^{i,\ell}$ and we make it adjacent to both $s_{i,j}$ and $s_{j,\ell}$. We also add to H a new vertex $v_1^{0,2}$ adjacent to $s_{1,2}$, a new vertex $v_1^{3,0}$ adjacent to $s_{3,1}$, and a new vertex r adjacent to $v_1^{0,2}$, which will be the *root* of H. Note that the graph constructed so far is a simple path P on $2\binom{k}{2} + 2$ vertices. We say that the vertices of the form $v_j^{i,\ell}$ are *occurrences* of vertex $v_j \in V(K_k)$. For $2 \leq j \leq k$, we add an edge between the root r and the first occurrence of vertex v_j in P (note that the edge between r and the first occurrence of v_1 already exists).

The first k selector vertices, namely $s_{1,2}, s_{2,3}, \dots, s_{k-1,k}, s_{k,1}$ will play a special role that will become clear later. To this end, for $1 \leq i \leq k$, we add an edge

[1] This assumption is not crucial for the construction, but helps in making it conceptually and notationally easier.

between the selector vertex $s_{i,i \pmod k)+1}$ and each of the occurrences of v_i that appear after $s_{i,i \pmod k)+1}$ in P. These edges will be called the *jumping edges* of H.

Let us denote by F the graph constructed so far. Finally, in order to construct H, we replace each vertex of the form $v_j^{i,\ell}$ in F with a whole copy of the vertex set V_j of G and make each of these new vertices adjacent to all the neighbors of $v_j^{i,\ell}$ in F. This completes the construction of H. Note that $\mathbf{td}(H) \leq \binom{k}{2} + 1$, as the removal of the $\binom{k}{2}$ selector vertices from H results in a star centered at r and isolated vertices.

We now proceed to describe the color palette X, the coloring function χ, and the cost function cc, which altogether will encode the edges of G and will ensure the desired properties of the reduction. For simplicity, we associate a distinct color with each edge of H, and thus, with slight abuse of notation, it is enough to describe the cost function cc for every ordered pair of incident edges of H. We will use just three different costs: 0, 1, and B, where B can be set as any real number strictly greater than $\binom{k}{2}$. For each ordered pair of incident edges e_1, e_2 of H, we define

$$cc(e_1, e_2) = \begin{cases} 0, & \text{if } e_1 = \{\hat{x}, s_{i,j}\} \text{ and } e_2 = \{s_{i,j}, \hat{y}\} \text{ is a jumping edge such that} \\ & \hat{x}, \hat{y} \text{ are copies of vertices } x, y \in V_i, \text{ respectively, with } x \neq y, \text{ or} \\ & \text{if } e_1 = \{r, \hat{x}\} \text{ and } e_2 = \{\hat{x}, s_{1,2}\}, \text{ where } \hat{x} \text{ is a copy of a vertex} \\ & x \in V_1, \text{ or} \\ & \text{if } e_1 \text{ and } e_2 \text{ are the two edges that connect a vertex in a copy} \\ & \text{of a color class } V_i \text{ to a selector vertex.} \\ \\ 1, & \text{if } e_1 = \{\hat{x}, s_{i,j}\} \text{ and } e_2 = \{s_{i,j}, \hat{y}\}, \text{ where } \hat{x} \text{ is a copy of a vertex} \\ & x \in V_i \text{ and } \hat{y} \text{ is a copy of a vertex } y \in V_j \text{ such that } \{x, y\} \in E(G). \\ \\ B, & \text{otherwise.} \end{cases}$$

This completes the construction of (H, X, χ, r, cc), which can be clearly performed in polynomial time.

Claim 1 [$\star$]. *The average degree of H is bounded by* 8.

We now claim that H contains and arborescence T rooted at r with cost at most $\binom{k}{2}$ if and only if G contains a multicolored k-clique[2]. Note that the simple path P described above naturally defines a partial left-to-right ordering among the vertices of H, and hence any arborescence rooted at r contains *forward* and *backward* edges defined in an unambiguous way. Note also that all costs that involve a backward edge are equal to B, and therefore no such edge can be contained in an arborescence of cost at most $\binom{k}{2}$.

[2] If the costs associated with colors are restricted to be strictly positive, we can just replace cost 0 with cost ε, for an arbitrarily small positive real number ε, and ask for an arborescence in H of cost strictly smaller than $\binom{k}{2} + 1$.

Suppose first that G contains a multicolored k-clique with vertices $v_1, v_2, \ldots, v_k$, where $v_i \in V_i$ for $1 \leq i \leq k$. Then we define the edges of the spanning tree T of H as follows. Tree T contains the edges of a left-to-right path Q that starts at the root r, contains all $\binom{k}{2}$ selector vertices and connects them, in each occurrence of a set V_i, to the copy of vertex v_i defined by the k-clique. Since in Q the selector vertices connect copies of pairwise adjacent vertices of G, the cost incurred so far by T is exactly $\binom{k}{2}$. For $1 \leq i \leq k$, we add to Q the edges from r to all vertices in the first occurrence of V_i that are not contained in Q. Note that the addition of these edges to T incurs no additional cost. Finally, we will use the jumping edges to reach the uncovered vertices of H. Namely, for $1 \leq i \leq k$, we add to T an edge between the selector vertex $s_{i,i \pmod k)+1}$ and all occurrences of the vertices in V_i distinct from v_i that appear after $s_{i,i \pmod k)+1}$. Note that since the jumping edges in T contain copies of vertices distinct from the the ones in the k-clique, these edges incur no additional cost either. Therefore, $cc(T, r) = \binom{k}{2}$, as we wanted to prove.

Conversely, suppose now that H has an arborescence T rooted at r with cost at most $\binom{k}{2}$. Clearly, all costs incurred by the edges in T are either 0 or 1. For a selector vertex $s_{i,j}$, we call the edges joining $s_{i,j}$ to the vertices in the occurrence of V_i right before $s_{i,j}$ (resp., in the occurrence of V_j right after $s_{i,j}$) the *left* (resp., *right*) edges of this selector vertex.

Claim 2 [$\star$]**.** *Tree T contains exactly one left edge and exactly one right edge of each selector vertex of H.*

By Claim 2, tree T contains a path Q' that chooses exactly one vertex from each occurrence of a color class of G. We shall now prove that, thanks to the jumping edges, these choices are *coherent*, which will allow us to extract the desired multicolored k-clique in G.

Claim 3. *For every $1 \leq i \leq k$, the vertices in the copies of color class V_i contained in Q' all correspond to the same vertex of G, denoted by v_i.*

Proof. Assume for contradiction that for some index i, the vertices in the copies of color class V_i contained in Q' correspond to at least two distinct vertices v_i and v'_i of G, in such a way that v_i is the selected vertex in the first occurrence of V_i, and v'_i occurs later, say in the jth occurrence of V_i. Therefore, the copy of v_i in the jth occurrence of V_i does not belong to path Q', so for this vertex to be contained in T, by construction it is necessarily an endpoint of a jumping edge e starting at the selector vertex $s_{i,i \pmod k)+1}$. But then the cost incurred in T by the edges e' and e, where e' is the edge joining the copy of v_i in the first occurrence of V_i to the selector vertex $s_{i,i \pmod k)+1}$, equals B, contradicting the hypothesis that $cc(T, r) \leq \binom{k}{2}$. □

Finally, we claim that the vertices $v_1, v_2, \ldots, v_k$ defined by Claim 3 induce a multicolored k-clique in G. Indeed, assume for contradiction that there exist two such vertices v_i and v_j such that $\{v_i, v_j\} \notin E(G)$. Then the cost in T incurred by the two edges connecting the copies of v_i and v_j to the selector vertex $s_{i,j}$

(by Claim 2, these two edges indeed belong to T) would be equal to B, contracting again the hypothesis that $cc(T, r) \leq \binom{k}{2}$. This concludes the proof of the theorem. $\square$

In the next theorem we prove that the MinCCA problem is W[1]-hard on multigraphs parameterized by the tree-cutwidth of the input graph. Note that this result does not imply Theorem 1, which applies to graphs without multiple edges.

Theorem 2 [$\star$]. *The* MinCCA *problem is* W[1]*-hard on multigraphs parameterized by the tree-cutwidth of the input multigraph.*

3.2 NP-hardness on Planar Graphs

In this subsection we prove that the MinCCA problem remains NP-hard on planar graphs. In order to prove this result, we need to introduce the Planar Monotone 3-sat problem. An instance of 3-sat is called *monotone* if each clause is monotone, that is, each clause consists only of positive variables or only of negative variables. We call a clause with only positive (resp., negative) variables a *positive* (resp., *negative*) clause. Given an instance ϕ of 3-sat, we define the bipartite graph G_ϕ that has one vertex per each variable and each clause, and has an edge between a variable-vertex and a clause-vertex if and only if the variable appears (positively or negatively) in the clause. A *monotone rectilinear representation* of a monotone 3-sat instance ϕ is a *planar* drawing of G_ϕ such that all variable-vertices lie on a path, all positive clause-vertices lie *above* the path, and all negative clause-vertices lie *below* the path. In the Planar Monotone 3-sat problem, we are given a monotone rectilinear representation of a planar monotone 3-sat instance ϕ, and the objective is to determine whether ϕ is satisfiable. Berg and Khosravi [4] proved that the Planar Monotone 3-sat problem is NP-complete.

Theorem 3. *The* MinCCA *problem is* NP*-hard on planar graphs even when restricted to instances with at most 6 colors and 0/1 symmetric costs.*

Proof. We reduce from the Planar Monotone 3-sat problem. Given a monotone rectilinear representation of a planar monotone 3-sat instance ϕ, we build an instance (H, X, χ, r, f) of MinCCA as follows. We denote the variable-vertices of G_ϕ as $\{x_1, \ldots, x_n\}$ and the clause-vertices of G_ϕ as $\{C_1, \ldots, C_m\}$. Without loss of generality, we assume that the variable-vertices appear in the order $x_1, \ldots, x_n$ on the path P of G_ϕ that links the variable-vertices. For every variable-vertex x_i of G_ϕ, we add to H a gadget consisting of four vertices $x_i^\ell, x_i^r, x_i^+, x_i^-$ and five edges $\{x_i^\ell, x_i^+\}$, $\{x_i^+, x_i^r\}$, $\{x_i^r, x_i^-\}$, $\{x_i^-, x_i^\ell\}$, $\{x_i^+, x_i^-\}$. We add to H a new vertex r, which we set as the root, and we add the edge $\{r, x_1^\ell\}$. For every $i \in \{1, \ldots, n-1\}$, we add to H the edge $\{x_i^r, x_{i+1}^\ell\}$. We add to H all clause-vertices $C_1, \ldots, C_m$. For every $i \in \{1, \ldots, n\}$, we add an edge between vertex x_i^+ and each clause-vertex of G_ϕ in which variable x_i appears positively, and an edge between vertex x_i^- and each clause-vertex of G_ϕ in which variable x_i appears negatively. This completes the construction of H. Since G_ϕ is planar and all positive (resp., negative) clause-vertices appear above (resp., below) the path P, it is easy to see that the graph H is planar as well.

We define the color palette as $X = \{1, 2, 3, 4, 5, 6\}$. Let us now describe the edge-coloring function χ. For every clause-vertex C_j, we color arbitrarily its three incident edges with the colors $\{4, 5, 6\}$, so that each edge incident to C_j gets a different color. For every $i \in \{1, \dots, n\}$, we define $\chi(\{x_i^\ell, x_i^+\}) = \chi(\{x_i^r, x_i^-\}) = 1$, $\chi(\{x_i^+, x_i^r\}) = \chi(\{x_i^-, x_i^\ell\}) = 2$, and $\chi(\{x_i^+, x_i^-\}) = 3$. We set $\chi(\{r, x_1^\ell\}) = 4$ and for every $i \in \{1, \dots, n-1\}$, $\chi(\{x_i^r, x_{i+1}^\ell\}) = 4$. Finally, we define the cost function cc to be symmetric and, for every $i \in \{1, 2, 3, 4, 5, 6\}$, we set $cc(i, i) = 0$. We define $cc(1, 2) = 1$ and $cc(1, 3) = cc(2, 3) = 0$. For every $i \in \{4, 5, 6\}$, we set $cc(1, i) = cc(2, i) = 0$ and $cc(3, i) = 1$. Finally, for every $i, j \in \{4, 5, 6\}$ with $i \neq j$ we set $cc(i, j) = 1$. The following claim concludes the proof.

Claim 4 [$\star$]. *H contains an arborescence T rooted at r with cost 0 if and only if the formula ϕ is satisfiable.* $\square$

Note that the above proof actually implies that MINCCA cannot be approximated to *any* positive ratio on planar graphs in polynomial time, since an optimal solution has cost 0. We do not know whether such a strong inapproximability result holds even if we do not allow to use costs 0 among different colors.

In the next theorem we present a modification of the previous reduction showing that the MINCCA problem remains hard even if the maximum degree of the input planar graph is bounded.

Theorem 4 [$\star$]. *The* MINCCA *problem is* NP*-hard on planar graphs even when restricted to instances with at most 8 colors, maximum degree bounded by* 4*, and 0/1 symmetric costs.*

4 Conclusions and Further Research

In this article we proved several hardness results for the MINCCA problem. In particular, we proved that the problem is W[1]-hard parameterized by treewidth on general graphs, and that it is NP-hard on planar graphs, but we do not know whether it is W[1]-hard parameterized by treewidth (or treedepth) on planar graphs.

On the other hand, we provided an FPT algorithm for a restricted version of tree-cutwidth, and we proved that the problem is W[1]-hard on multigraphs parameterized by tree-cutwidth. While we were not able to prove this W[1]-hardness result on graphs without multiple edges, we believe that it is indeed the case. It would be natural to consider other structural parameters such as the size of a vertex cover or a feedback vertex set.

Finally, it would be interesting to try to generalize our techniques to prove hardness results or to provide efficient algorithms for other reload cost problems that have been studied in the literature [6,8,10,19].

Acknowledgment. We would like to thank the anonymous referees for helpful comments that improved the presentation of the manuscript

References

1. Amaldi, E., Galbiati, G., Maffioli, F.: On minimum reload cost paths, tours, and flows. Networks **57**(3), 254–260 (2011)
2. Arkoulis, S., Anifantis, E., Karyotis, V., Papavassiliou, S., Mitrou, N.: On the optimal, fair and channel-aware cognitive radio network reconfiguration. Comput. Netw. **57**(8), 1739–1757 (2013)
3. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Switzerland (2015)
4. de Berg, M., Khosravi, A.: Optimal binary space partitions for segments in the plane. Int. J. Comput. Geom. Appl. **22**(3), 187–206 (2012)
5. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, London (2013)
6. Çelenlioğlu, M. R., Gözüpek, D., Mantar, H. A.: A survey on the energy efficiency of vertical handover mechanisms. In: Proceedings of the International Conference on Wireless and Mobile Networks (WiMoN) (2013)
7. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. Springer, Heidelberg (2006)
8. Galbiati, G.: The complexity of a minimum reload cost diameter problem. Discrete Appl. Math. **156**(18), 3494–3497 (2008)
9. Galbiati, G., Gualandi, S., Maffioli, F.: On minimum changeover cost arborescences. In: Pardalos, P.M., Rebennack, S. (eds.) SEA 2011. LNCS, vol. 6630, pp. 112–123. Springer, Heidelberg (2011)
10. Galbiati, G., Gualandi, S., Maffioli, F.: On minimum reload cost cycle cover. Discrete Appl. Math. **164**, 112–120 (2014)
11. Ganian, R., Kim, E.J., Szeider, S.: Algorithmic applications of tree-cut width. In: Italiano, G.F., Pighizzini, G., Sannella, D.T. (eds.) MFCS 2015. LNCS, vol. 9235, pp. 348–360. Springer, Heidelberg (2015)
12. Gourvès, L., Lyra, A., Martinhon, C., Monnot, J.: The minimum reload s-t path, trail and walk problems. Discrete Appl. Math. **158**(13), 1404–1417 (2010)
13. Gozupek, D., Buhari, S., Alagoz, F.: A spectrum switching delay-aware scheduling algorithm for centralized cognitive radio networks. IEEE Trans. Mobile Comput. **12**(7), 1270–1280 (2013)
14. Gözüpek, D., Shachnai, H., Shalom, M., Zaks, S.: Constructing minimum changeover cost arborescenses in bounded treewidth graphs. Theorerical Comput. Sci. **621**, 22–36 (2016)
15. Gözüpek, D., Shalom, M., Voloshin, A., Zaks, S.: On the complexity of constructing minimum changeover cost arborescences. Theorerical Comput. Sci. **540**, 40–52 (2014)
16. Kim, E., Oum, S., Paul, C., Sau, I., Thilikos, D.M.: An FPT 2-approximation for tree-cut decomposition. In: Sanità, L., et al. (eds.) WAOA 2015. LNCS, vol. 9499, pp. 35–46. Springer, Heidelberg (2015). doi:10.1007/978-3-319-28684-6_4
17. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms, vol. 31. Oxford University Press, Oxford (2006)
18. Pietrzak, K.: On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. J. Comput. Syst. Sci. **67**(4), 757–771 (2003)
19. Wirth, H.-C., Steffan, J.: Reload cost problems: minimum diameter spanning tree. Discrete Appl. Math. **113**(1), 73–85 (2001)
20. Wollan, P.: The structure of graphs not admitting a fixed immersion. J. Comb. Theor. Ser. B **110**, 47–66 (2015)