Theoretical Computer Science 690 (2017) 91–103

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Parameterized complexity of the MINCCA problem on graphs of bounded decomposability ☆

Didem Gözüpek [a], Sibel Özkan [b], Christophe Paul [c], Ignasi Sau [c,*], Mordechai Shalom [d,e,1]

[a] Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey
[b] Department of Mathematics, Gebze Technical University, Kocaeli, Turkey
[c] CNRS, LIRMM, Université de Montpellier, Montpellier, France
[d] TelHai College, Upper Galilee, 12210, Israel
[e] Department of Industrial Engineering, Boğaziçi University, Istanbul, Turkey

ARTICLE INFO

*Article history:*
Received 3 August 2016
Received in revised form 19 May 2017
Accepted 19 June 2017
Available online 27 June 2017
Communicated by F.V. Fomin

*Keywords:*
Minimum changeover cost arborescence
Parameterized complexity
FPT algorithm
Treewidth
Tree-cutwidth
Planar graph

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. (2016) that the MinCCA problem when parameterized by the treewidth and the maximum degree of the input graph is in FPT. In this article we present the following hardness results for MinCCA:

- the problem is W[1]-hard when parameterized by the vertex cover number of the input graph, even on graphs of degeneracy at most 3. In particular, it is W[1]-hard parameterized by the treewidth of the input graph, which answers the main open problem in the work of Gözüpek et al. (2016);
- it is W[1]-hard on multigraphs parameterized by the tree-cutwidth of the input multigraph; and
- 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.



☆ An extended abstract of this article appeared in the *Proceedings of the 42nd International Workshop on Graph-Theoretic Concepts in Computer Science (**WG**), volume 9941 of LNCS, pages 195–206, Istanbul, Turkey, June **2016***. This work was supported by the bilateral research program of CNRS and TUBITAK under grant no. 114E731.

* Corresponding author.
*E-mail addresses:* didem.gozupek@gtu.edu.tr (D. Gözüpek), s.ozkan@gtu.edu.tr (S. Özkan), paul@lirmm.fr (C. Paul), sau@lirmm.fr (I. Sau), cmshalom@telhai.ac.il (M. Shalom).

[1] The work of this author is supported in part by the TUBITAK 2221 Programme.

http://dx.doi.org/10.1016/j.tcs.2017.06.013
0304-3975/

## 1. Introduction

Numerous network optimization problems can be modeled by edge-colored graphs. A path in an edge-colored graph may incur at every internal vertex a cost that depends on the colors of the two incident edges. In the literature, this cost is referred to as *reload cost* or *changeover cost*. 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 by Gözüpek et al. [14].

In heterogeneous telecommunications networks, 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 [5], 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 [1,13]. 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, in some cases, outweighs even the cost of transporting the cargo from one place to another using a single mode of transportation [20]. 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 by Wirth and Steffan [20] that considered the problem of finding an arborescence having minimum diameter with respect to reload cost. The same problem was considered later by Galbiati et al. [8]. The work of Galbiati et al. [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. Gourvès et al. [12] studied the problems of finding a path, trail or walk connecting two given vertices with minimum total reload cost.

In their work [9], Galbiati et al. introduced the *Minimum Changeover Cost Arborescence* (MinCCA) problem which is the focus of this work. Given a root vertex, the MinCCA problem consists in finding 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 of Gözüpek et al. [15] investigated several special cases of the problem such as bounded cost values, bounded degree, and bounded number of colors. In that work inapproximability results as well as a polynomial-time algorithms and approximation algorithms are presented for special cases.

In this paper, we study the MinCCA problem from the perspective of parameterized complexity; see the books [2,4,7,19] for an introduction to the domain. 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* 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. The class of all fixed-parameter tractable problems is denoted as FPT. 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 in FPT.

The parameterized complexity of reload cost problems is largely unexplored in the literature. To the best of our knowledge, the work of Gözüpek et al. [14] is the only one that focuses on this issue by studying the MinCCA problem on bounded treewidth graphs. In particular, Gözüpek et al. [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. We would like to note that these parameters have practical importance in communication networks. Indeed, for instance, many networks that model real-life situations appear to have small treewidth [16,18].

In this article we prove that the MinCCA problem is W[1]-hard parameterized by the vertex cover number of the input graph, even on graphs of degeneracy at most 3. In particular, it is W[1]-hard parameterized by the treewidth of the input graph. This answers the main open issue pointed out by Gözüpek et al. [14], and is also interesting since most problems are known to be in FPT when parameterized by the treewidth of their input graph.

In view of the above results, it makes sense to study the parameterized complexity of the MinCCA problem for parameters that lie in between treewidth and treewidth plus maximum degree. A natural candidate is tree-cut width (see Fig. 1), a width parameter recently introduced by Wollan [21] that plays a fundamental role in the structure of graphs not admitting a fixed immersion (see Section 2 for the precise definition). In this direction, we prove that MinCCA is W[1]-hard on multigraphs parameterized by the tree-cutwidth of the input multigraph.

We also prove that MinCCA 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 Section 2 we introduce some basic definitions and preliminaries, as well as a formal definition of the MinCCA problem. Our main result is in Section 3, where we prove that the problem is W[1]-hard parameterized by the vertex cover number of the input graph, even if the input graph has degeneracy at most 3. In Section 4 we prove that the problem is W[1]-hard on *multigraphs* parameterized by the tree-cutwidth of the input graph. In Section 5 we prove that the problem remains NP-hard on planar graphs. Finally, Section 6 concludes the paper.

## 2. Preliminaries

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 $G$ and a subset $U \subseteq V(G)$ of the vertices of $G$, $\delta_G(U) := \left\{\{u, u'\} \in E(G) \mid u \in U, u' \notin U\right\}$ 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 $G$. We denote by $N_G(U)$ the *open 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 $U$. 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 $U$. For a subset $U$ of vertices of $G$, and a subset $F$ of its edges we denote by $G[U \cup F]$ the graph induced by these vertices and edges, that is obtained by adding to $G[U]$ the edges of $F$ and their endpoints. Formally, $G[U \cup F] := (U \cup V(F), F \cup E(G) \cap U \times U)$. The *degeneracy* of a graph $G$ is the smallest number $k$ such that every induced subgraph of $G$ has a vertex of degree at most $k$. A *vertex cover* of a graph $G$ is a subset $S \subseteq V(G)$ such that $G - S$ is an independent set. The minimum cardinality of a vertex cover of a graph $G$ is called the *vertex cover number* of $G$ and denoted by $\mathbf{vc}(G)$.

A digraph $T$ is a *rooted tree* or *arborescence* if its underlying graph is a tree and it contains a *root* vertex with a directed path from every other vertex to it. Every non-root vertex $v$ of $T$ 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 and treewidth* 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 $G$.

*Tree-cutwidth* We now explain the concept of tree-cutwidth and follow the notation of Ganian et al. [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 $G_t = G[Y_t]$, and by $\mathsf{cut}(t) = \delta(Y_t)$ 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 $Y_0 = V(G) \setminus (X_t \cup \bigcup_{i=1}^{\ell} Y_i)$. We first contract each set $Y_i$ to a single vertex $y_i$ for every $i \in [0, \ell]$ 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 $\max\{\mathsf{adh}(t), \mathsf{tor}(t) \mid t \in V(T)\}$. 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 $G$.

Fig. 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 results (cf. [11,17,21]). Furthermore, the vertex cover number and tree-cutwidth are not related to each other, i.e., in general none of them bounds the other one.

*Reload and changeover costs* We follow the notation and terminology of Wirth and Steffan [20] 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 $\chi : E(G) \to X$ is the *coloring function*. Given a coloring function $\chi$, and a color $x \in X$, we denote by $E_x^{\chi}$, 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 $\mathsf{cc} : X^2 \to \mathbb{N}_0$ satisfying

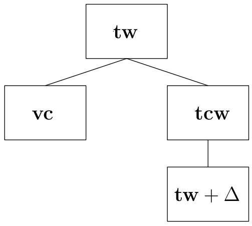


**Fig. 1.** Relationships between the graph parameters considered in this paper. In the figure, $A$ being a child of $B$ (drawn beneath $B$) means that every graph class with bounded $A$ has also bounded $B$, but the converse is not necessarily true [11].

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

The cost of traversing two incident edges $e_1, e_2$ is $\mathrm{cc}(e_1, e_2) := \mathrm{cc}(\chi(e_1), \chi(e_2))$.

We say that an instance satisfies the *triangle inequality*, if (in addition to the above) the cost function satisfies $\mathrm{cc}(e_1, e_3) \leq \mathrm{cc}(e_1, e_2) + \mathrm{cc}(e_2, e_3)$ whenever $e_1, e_2$ and $e_3$ are incident to the same vertex.

The changeover cost of a path $P$ of length $\ell \geq 2$ with edges $e_1, e_2, \ldots, e_\ell$ is $\mathrm{cc}(P) := \sum_{i=2}^{\ell} \mathrm{cc}(e_{i-1}, e_i)$. We define $\mathrm{cc}(P) = 0$ 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 $\mathrm{cc}(T, r) := \sum_{e \in E(T)} \mathrm{cc}(prev(e), e)$. When there is no ambiguity about the vertex $r$, we denote $\mathrm{cc}(T, r)$ by $\mathrm{cc}(T)$.

*Statement of the problem* As defined by Galbiati et al. [9], the MinCCA problem aims to find an arborescence rooted at $r$ with minimum changeover cost. 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 $\mathrm{cc} : X^2 \to \mathbb{N}_0$.
> **Output:** An arborescence $T$ of $G$ minimizing $\mathrm{cc}(T, r)$.

## 3. W[1]-hardness with parameter vertex cover

Before stating our main result, we need to define the following parameterized problem.

> Multicolored $k$-Clique
> **Input:** A graph $G$, a coloring function $c : V(G) \to \{1, \ldots, k\}$, 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 [6], and therefore we may make this assumption as well.

**Theorem 1.** *The* MinCCA *problem is* W*[1]-hard when parameterized either by the vertex cover number of the input graph, even when its degeneracy is 3.*

**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 $c' : V(G') \to \{1, \ldots, k+1\}$ 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 $(H, X, \chi, r, \mathrm{cc})$ 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 Eulerian circuit of the complete graph $K_k$ on the vertex set $V(K_k) = \{v_1, \ldots, v_k\}$ that starts by visiting, in this order, the vertices

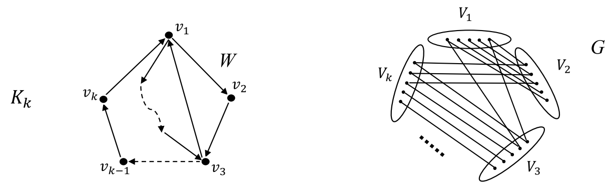


**Fig. 2.** The complete graph $K_k$ and an Eulerian circuit $W$ in $K_k$ starting with $v_1, v_2, \ldots, v_k, v_1$ and ending with $v_3, v_1$. A $k$-colored graph $G$ is also illustrated.

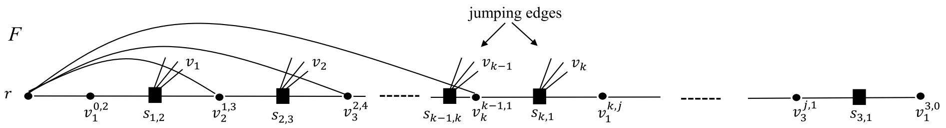


**Fig. 3.** The graph $F$.

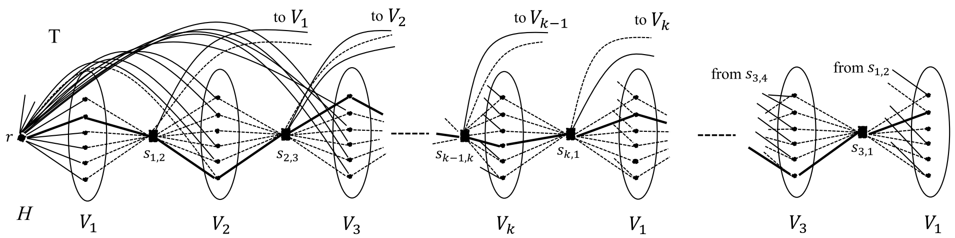


**Fig. 4.** The graph $H$ and a solution arborescence $T$ drawn in solid lines. The path $Q$ is drawn in thicker lines.

$v_1, v_2, \ldots, v_k, v_1$, and ends by visiting $v_3$ and finally $v_1$[2]; see Fig. 2. Note that $W$ always exists since whenever $k > 3$ we can construct the $k+1$ cycle $v_1 v_2 \cdots v_k v_3 v_1$ and combine it with any Eulerian circuit of the remaining graph to get $W$. 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 $2 \le j \le k$, 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 $s_{1,2}, s_{2,3}, \ldots, s_{k-1,k}, s_{k,1}$ will play a special role that will become apparent later. To this end, for $1 \le i \le k$, we add an edge 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; see Fig. 3. 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 $H$; see Fig. 4.

At this point we note that the vertex cover number of $H$ is at most $\binom{k}{2}+1$, since the selector vertices and $r$ constitute a vertex cover of $H$. The degeneracy of $H$ is 3 since every induced subgraph of $H$ that contains a non-selector vertex $v$ has minimum degree at most 3, and every induced subgraph that does not contain such a vertex has minimum degree 0.

We now proceed to describe the color palette $X$, the coloring function $\chi$, 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 $B = \binom{k}{2}+1$. For each ordered pair of incident edges $e_1, e_2$ of $H$, we define

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

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

This completes the construction of $(H, X, \chi, r, \mathrm{cc})$, which can be clearly performed in polynomial time.[3]

We now claim that $H$ contains an arborescence $T$ rooted at $r$ with cost at most $\binom{k}{2}$ if and only if $G$ contains a multicolored $k$-clique. 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 $\binom{k}{2}$.

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 $1 \le i \le k$. Then we define the edges of the arborescence $T$ of $H$ as follows. The 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 $1 \le i \le k$, 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 $1 \le i \le k$, 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 $s_{i,i \pmod k+1}$; see the solid edges in Fig. 4. Note that since the jumping edges in $T$ contain copies of vertices distinct from the ones in the $k$-clique, these edges incur no additional cost either. Therefore, $\mathrm{cc}(T, r) = \binom{k}{2}$, 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 1.** *The tree $T$ contains exactly one left edge and exactly one right edge of each selector vertex of $H$.*

**Proof:** Since only forward edges are allowed in $T$, and $T$ should be a tree, clearly for each selector vertex exactly one of its left edges belongs to $T$. Thus, it just remains to prove that $T$ contains exactly one right edge of each selector vertex.

Let $s_{i,j}$ and $s_{j,\ell}$ be two consecutive selector vertices. Let $e$ be the left edge of $s_{j,\ell}$ in $T$ and let $v_j$ be the vertex of the copy of $V_j$ contained in $e$. Again, since backward edges are not allowed in $T$, $v_j$ needs to be incident with another forward edge $e'$ of $T$. If this edge $e'$ contains $r$ or if it is a jumping edge, then the cost incurred by $T$ during the traversal from $e'$ to $e$ would be equal to $B$, a contradiction to the assumption that $\mathrm{cc}(T, r) \le \binom{k}{2} < B$. Therefore, $e'$ is necessarily one of the right edges of $s_{i,j}$, so at least one of the right edges of the selector vertex $s_{i,j}$ belongs to $T$.

As for the right edges of the last selector vertex, namely $s_{3,1}$, if none of them belonged to $T$, then there would be a jumping edge going to the last copy of $V_1$ such that, together with the left edge of the selector vertex $s_{1,2}$ that belongs to $T$, would incur a cost of $B$, which is impossible.

We have already proved that exactly one left edge and *at least* one of the right edges of each selector vertex belong to $T$. For each selector vertex $s_{i,j}$, its left edge in $T$ together with *each* of its right edges in $T$ incur at cost of at least 1. But as there are $\binom{k}{2}$ selector vertices in $H$, and by hypothesis the cost of $T$ is at most $\binom{k}{2}$, we conclude that exactly one of the right edges of each selector vertex belongs to $T$, as we wanted to prove. $\diamond$

By Claim 1, the 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 consistent, i.e., copies of the same vertex. This will allow us to extract the desired multicolored $k$-clique in $G$.

**Claim 2.** *For every $1 \le i \le k$, the vertices in the copies of color class $V_i$ contained in $Q'$ all correspond to the same vertex of $G$, denoted by $v_i$.*

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

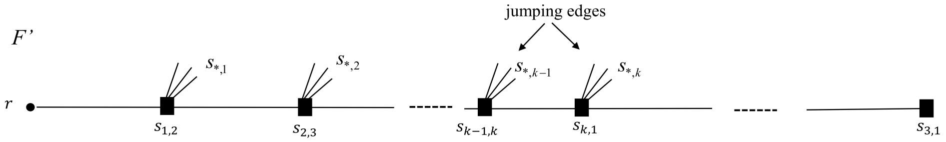


**Fig. 5.** The graph $F'$.

**Proof:** Assume for a 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 $j$th occurrence of $V_i$. Therefore, the copy of $v_i'$ in the $j$th 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 assumption $\mathrm{cc}(T,r) < B$. ◇

Finally, we claim that the vertices $v_1, v_2, \ldots, v_k$ defined by Claim 2 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 1, these two edges indeed belong to $T$) would be equal to $B$, contracting again the assumption $\mathrm{cc}(T,r) < B$. □

## 4. W[1]-hardness on multigraphs with parameter tree-cut width

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

**Proof.** As in Theorem 1, we reduce again from Multicolored $k$-Clique. Given an instance $(G, c, k)$ of Multicolored $k$-Clique with $k$ odd, we proceed to construct an instance $(H, X, \chi, r, \mathrm{cc})$ of MinCCA. The first steps of the construction are similar to those of Theorem 1. Namely, let $F$ be the graph constructed in the proof of Theorem 1 (see Fig. 3), and let $F'$ be the graph constructed from $F$ as follows (see Fig. 5). We delete the last vertex of $F$, namely $v_1^{3,0}$, and all edges incident with the root $r$ except the edge $\{r, v_1^{0,2}\}$. Finally, for every vertex of $F'$ of the form $v_\ell^{i,j}$ (that is, a vertex that is neither the root nor a selector vertex), let $e_1$ and $e_2$ be the two edges of the path $P$ incident with $v_\ell^{i,j}$, such that $e_1$ is to the left of $e_2$. Then we contract the edge $e_2$, and we give to the newly created vertex the name of the selector vertex incident with $e_2$. This completes the construction of $F'$. Note that $|V(F')| = \binom{k}{2} + 1$. Finally, in order to construct $H$, we proceed as follows. For every edge $e$ of $F'$ which is not a jumping edge, let $s_{i,j}$ be its right endpoint. Then we replace $e$ with a multiedge with multiplicity $|V_i|$, and we associate each of these edges with a distinct vertex in $V_i \subseteq V(G)$. These edges are called the *horizontal* edges of $H$. On the other hand, for every $1 \le i \le k$, and for every jumping edge $e$ whose left endpoint is the selector vertex $s_{i,i \pmod k)+1}$, we replace $e$ with a multiedge with multiplicity $|V_i|$, and we subdivide each of these new edges once. Each of these new vertices $\hat{x}$ is associated with a distinct vertex in $V_i \subseteq V(G)$. Let us call the selector vertices of the form $s_{i,i \pmod k)+1}$ *special* selector vertices. This completes the construction of $H$.

**Claim 3.** *The tree-cutwidth of H is at most* $\binom{k}{2} + 1$.

**Proof:** We proceed to construct a tree-cut decomposition $(T, \mathcal{X})$ of $H$ of width at most $\binom{k}{2} + 1$. Let $T$ be a star with $|V(H)| - \binom{k}{2} - 1$ leaves rooted at its center. If $t$ is the center of this star $T$, then the bag $X_t$ contains the root $r$ of $H$ together with the $\binom{k}{2}$ selector vertices. If $t$ is a leaf of $T$, then the bag $X_t$ contains a single vertex, in such a way that each of the remaining $|V(H)| - \binom{k}{2} - 1$ vertices of $T$ is associated with one of the leaves. For every leaf $t \in V(T)$, it holds that $\mathrm{adh}(t) = 2$, as every vertex in $H$ that is neither the root nor a selector vertex has degree exactly 2. Also, for every leaf $t$ of $T$, clearly $\mathrm{tor}(t) \le 2$, as $|X_t| = 1$ and $t$ has degree 1 in $T$. Finally, if $t$ the root of $T$, then when considering the torso $H_t$, every vertex in a leaf-bag gets dissolved, as each such vertex has exactly 2 neighbors in $X_t$. Therefore, $\mathrm{tor}(t) \le \binom{k}{2} + 1$. ◇

We now proceed to describe the color palette $X$, the coloring function $\chi$, and the cost function cc, which altogether will encode the edges of $G$ and will ensure the desired properties of the reduction. For simplicity, as in the proof of Theorem 1, we again associate a distinct color with every edge of $H$, and thus, it is enough to describe the cost function cc for every ordered pair of incident edges of $H$. In this case, we will use just two different costs: 0 and 1. For every ordered pair of incident edges $e_1, e_2$ of $H$, we define

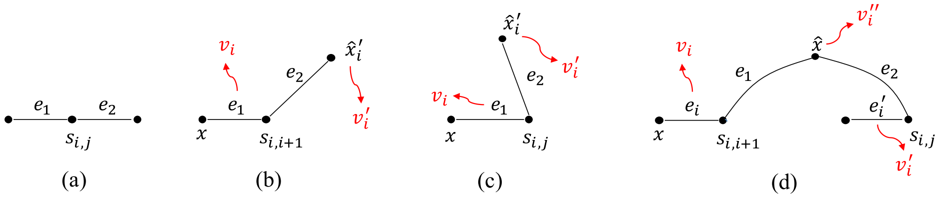


**Fig. 6.** (a)–(b)–(c) The three cases where $\mathrm{cc}(e_1, e_2) = 0$ in the proof of Theorem 2. (d) Construction in the proof. The vertices $v_i$, $v'_i$ and $v''_i$ correspond to the vertices in $V_i$ that are associated with the corresponding edges or vertices of $H$.

$$\mathrm{cc}(e_1, e_2) = \begin{cases} 0, & \text{if } e_1 = \{x, s_{i,j}\} \text{ and } e_2 = \{s_{i,j}, y\} \text{ are two horizontal edges} \\ & \text{such that } x \text{ is to the left of } y\text{, and the vertex in } V_i \text{ associated} \\ & \text{with } e_1 \text{ is adjacent in } G \text{ to the vertex in } V_j \text{ associated with } e_2\text{, or} \\ & \text{if } e_1 = \{x, s_{i,i \pmod k)+1}\} \text{ and } e_2 = \{s_{i,i \pmod k)+1}, \hat{x}'_i\} \text{ are} \\ & \text{such that } s_{i,i \pmod k)+1} \text{ is a special selector vertex, edge } e_1 \\ & \text{is horizontal and is associated with a vertex } v_i \in V_i\text{, edge } e_2 \\ & \text{arises from the subdivision of a jumping edge such that vertex} \\ & \hat{x}'_i \text{ is associated with a vertex } v'_i \in V_i \text{ with } v_i \neq v'_i\text{, or} \\ & \text{if } e_1 = \{x, s_{i,j}\} \text{ and } e_2 = \{s_{i,j}, \hat{x}'_i\} \text{ with } s_{i,j} \text{ being a selector} \\ & \text{vertex that is not special, edge } e_1 \text{ is horizontal and is associated} \\ & \text{with a vertex } v_i \in V_i\text{, edge } e_2 \text{ arises from the subdivision of a} \\ & \text{jumping edge such that vertex } \hat{x}'_i \text{ is associated with a vertex} \\ & v'_i \in V_i \text{ with } v_i = v'_i. \\ 1, & \text{otherwise.} \end{cases}$$

The three different cases above where $\mathrm{cc}(e_1, e_2) = 0$ are illustrated in Fig. 6(a)–(b)–(c), respectively. This completes the construction of $(H, X, \chi, r, \mathrm{cc})$, which can be clearly performed in polynomial time. We now claim that $H$ contains an arborescence $T$ rooted at $r$ with cost 0 if and only if $G$ contains a multicolored $k$-clique. Again, we assume that any arborescence in $H$ rooted at $r$ contains *forward* and *backward* edges defined in an unambiguous way.

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 arborescence $T$ of $H$ as follows. For each selector vertex $s_{i,j}$, $1 \leq i, j \leq k$, we add to $T$ its left horizontal edge associated with the vertex $v_i$ that belongs to the clique. For every jumping edge $\{s_{i,i \pmod k)+1}, s_{i,j}\}$ of $T'$, we do the following. Note that this edge has given rise to $|V_i|$ paths with two edges in $H$, and the vertices of $H$ in the middle of these paths, which we call *inner* vertices, are associated with the vertices in $V_i$. Then we add to $H$ a forward edge between $s_{i,i \pmod k)+1}$ and each inner vertex $\hat{x}$ associated with a vertex in $V_i$ distinct from the vertex $v_i$ that belongs to the clique. Note that $|V_i| - 1$ edges are added to $H$ in this way. Finally, we add a backward edge between $s_{i,j}$ and the inner vertex $\hat{x}$ associated with the vertex $v_i \in V_i$ that belongs to the clique. By the definition of the cost function and using the fact that the vertices $v_1, v_2, \ldots, v_k$ are pairwise adjacent in $G$, it can be easily checked that $\mathrm{cc}(T, r) = 0$, as we wanted to prove.

Conversely, suppose now that $H$ has an arborescence $T$ rooted at $r$ with cost 0. Clearly, all costs incurred by the edges in $T$ are necessarily 0. Since the cost incurred by the two edges incident with every inner vertex is equal to 1, necessarily $T$ contains a path $Q$ starting at the root $r$ and containing all $\binom{k}{2}$ selector vertices. Let $s_{i,j}$ be an arbitrary selector vertex distinct from the last one, and let $e_i$ and $e_j$ be its left and right incident horizontal edges in $Q$, respectively. Since $\mathrm{cc}(e_1, e_2) = 0$, necessarily the vertex $v_i \in V_i$ associated with $e_i$ is adjacent in $G$ to the vertex $v_j \in V_j$ associated with $e_j$. We say that the selector vertex $s_{i,j}$ has *selected* the vertex $v_i$. Our objective is to prove that these selections are *coherent*, in the sense that if two distinct selector vertices $s_{i,j}$ and $s_{i,\ell}$ have selected vertices $v_i$ and $v'_i$ in $V_i$, respectively, then $v_i = v'_i$. This property will be guaranteed by how the inner vertices are covered by $H$, as we proceed to prove.

By construction of $H$, each such inner vertex $\hat{x}$ is adjacent to a special selector vertex, say $s_{i,i+1}$, and to another selector vertex that is not special, say $s_{i,j}$. Let $e_1 = \{s_{i,i+1}, \hat{x}\}$ and let $e_2 = \{\hat{x}, s_{i,j}\}$. Note that either $e_1$ or $e_2$ belong to $T$, but not both, as otherwise these two edges would close a cycle with the path $Q$. Let also $e_i$ (resp., $e'_i$) be the edge that is to the left of $s_{i,i+1}$ (resp., $s_{i,j}$) in $Q$; see Fig. 6(d) for an illustration. Note that $e_i$ (resp., $e'_i$) is associated with a vertex $v_i \in V_i$ (resp., $v'_i \in V_i$), and similarly vertex $\hat{x}$ is associated with another vertex $v''_i \in V_i$. We distinguish two cases. Assume first that $v_i = v''_i$. In this case, by the definition of the cost function we have that $\mathrm{cc}(e_i, e_1) = 1$, so $e_1$ cannot belong to $T$, implying that $e_2$ belongs to $T$, which is possible only if $\mathrm{cc}(e'_i, e_2) = 1$, and this is true if and only if $v'_i = v_i$. This implies that the selections made by the selection vertices are coherent, as we wanted to prove. Otherwise, we have that $v_i \neq v''_i$. By the definition of the cost function, it holds that $\mathrm{cc}(e'_i, e_2) = 1$, and therefore, as we assume that $\mathrm{cc}(T, r) = 0$, necessarily $e_1$ belongs to $T$, which is indeed possible as $\mathrm{cc}(e_i, e_1) = 0$ because $v_i \neq v''_i$.

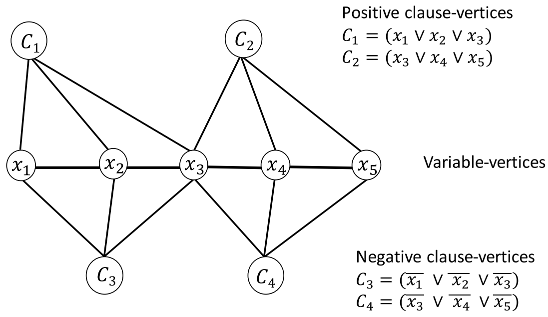


**Fig. 7.** A monotone rectilinear representation of a planar monotone 3-SAT instance.

By the above discussion, it follows that for each $1 \le i \le k$, all selector vertices of the form $s_{i,j}$, for every $1 \le j \le k$, $i \ne j$, have selected the same vertex $v_i \in V_i$. Furthermore, for every $1 \le i, j \le k$, $i \ne j$, it holds that $\{v_i, v_j\} \in E(G)$. That is, the selected vertices $v_1, v_2, \ldots, v_k$ induce a multicolored $k$-clique in $G$, concluding the proof of the theorem. $\square$

## 5. NP-hardness on planar graphs

In this section 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 $\phi$ of 3-SAT, we define the bipartite graph $G_\phi$ 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 $\phi$ is a *planar* drawing of $G_\phi$ 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; see Fig. 7 for an example.

In the Planar Monotone 3-SAT problem, we are given a monotone rectilinear representation of a planar monotone 3-SAT instance $\phi$, and the objective is to determine whether $\phi$ is satisfiable. Berg and Khosravi [3] 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 $\phi$, we build an instance $(H, X, \chi, r, f)$ of MinCCA as follows. The variable-vertices and the clause-vertices of $G_\phi$ are $\{x_1, \ldots, x_n\}$ and $\{C_1, \ldots, C_m\}$, respectively. Without loss of generality, we assume that the variable-vertices appear in the order $x_1, \ldots, x_n$ on the path $P$ of $G_\phi$ that links the variable-vertices. For every variable-vertex $x_i$ of $G_\phi$, 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 $G_\phi$ in which variable $x_i$ appears positively, and an edge between vertex $x_i^-$ and each clause-vertex of $G_\phi$ in which variable $x_i$ appears negatively. This completes the construction of $H$, which is illustrated in Fig. 8. Since $G_\phi$ 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 $\chi$. 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 $i \in \{1, \ldots, n\}$, 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 $i \in \{1, \ldots, n-1\}$, $\chi(\{x_i^r, x_{i+1}^\ell\}) = 4$. The function $\chi$ is also depicted in Fig. 8. Finally, we define the cost function cc to be symmetric and, for every $i \in \{1, 2, 3, 4, 5, 6\}$, we set $\mathrm{cc}(i, i) = 0$. We define $\mathrm{cc}(1, 2) = 1$ and $\mathrm{cc}(1, 3) = \mathrm{cc}(2, 3) = 0$. For every $i \in \{4, 5, 6\}$, we set $\mathrm{cc}(1, i) = \mathrm{cc}(2, i) = 0$ and $\mathrm{cc}(3, i) = 1$. Finally, for every $i, j \in \{4, 5, 6\}$ with $i \ne j$ we set $\mathrm{cc}(i, j) = 1$.

We now claim that $H$ contains an arborescence $T$ rooted at $r$ with cost 0 if and only if the formula $\phi$ is satisfiable.

Suppose first that $\phi$ is satisfiable, and fix a satisfying assignment of $\phi$. We proceed to define an arborescence $T$ rooted at $T$ with cost 0. $T$ contains the edge $\{r, x_1^\ell\}$ and, for every $i \in \{1, \ldots, n-1\}$, the edge $\{x_i^r, x_{i+1}^\ell\}$. For every $i \in \{1, \ldots, n\}$, if variable $x_i$ is set to 1 in the satisfying assignment of $\phi$, we add to $T$ the edges $\{x_i^\ell, x_i^+\}$, $\{x_i^+, x_i^-\}$, and $\{x_i^-, x_i^r\}$. Otherwise, if variable $x_i$ is set to 0 in the satisfying assignment of $\phi$, we add to $T$ the edges $\{x_i^\ell, x_i^-\}$, $\{x_i^-, x_i^+\}$, and $\{x_i^+, x_i^r\}$. Finally, for every $j \in \{1, \ldots, m\}$, let $x_t$ (resp., $\bar{x}_t$) be a literal in clause $C_j$ that is set to 1 by the satisfying assignment of $\phi$ (note that

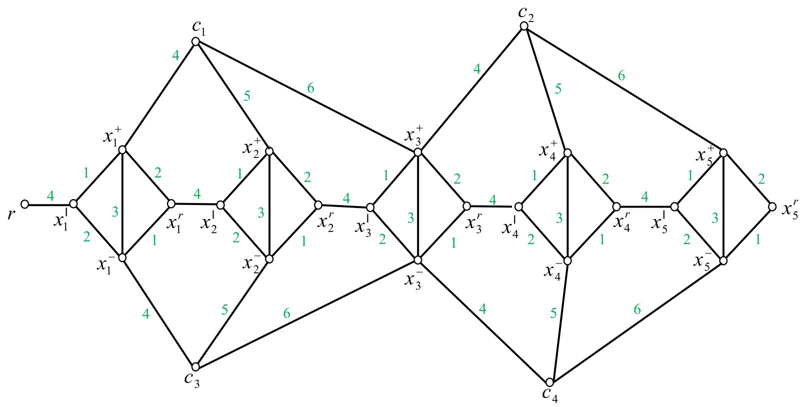


**Fig. 8.** The graph $H$ constructed by the reduction in Theorem 3 from the example of Fig. 7, together with the edge-coloring function $\chi$ (in green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for each clause we consider only *one* such literal). Then we add to $T$ an edge between vertex $C_j$ and vertex $x_t^+$ (resp., $x_t^-$). It can be easily checked that $T$ is an arborescence of $H$ with cost 0.

Conversely, suppose now that $H$ contains an arborescence $T$ rooted at $r$ with cost 0, and let us define a satisfying assignment of $\phi$. Since for every $i, j \in \{4, 5, 6\}$ with $i \neq j$ we have that $\mathtt{cc}(i, j) = 1$, for every clause-vertex $C_j$ exactly one of its incident edges belongs to $T$. From the structure of $H$ and from the fact that $\mathtt{cc}(1, 2) = 1$, it follows that in order for the tree $T$ to span all vertices of $H$, for every $i \in \{1, \ldots, n\}$ either the three edges $\{x_i^\ell, x_i^+\}$, $\{x_i^+, x_i^-\}$, $\{x_i^-, x_i^r\}$ or the three edges $\{x_i^\ell, x_i^-\}$, $\{x_i^-, x_i^+\}$, $\{x_i^+, x_i^r\}$ belong to $T$. In the former case, we set variable $x_i$ to 1, and in the latter case we set variable $x_i$ to 0. Since for every $i \in \{4, 5, 6\}$ we have that $\mathtt{cc}(3, i) = 1$, it follows that for every clause-vertex $C_j$, its incident edge that belongs to $T$ joins $C_j$ to a literal that is set to 1 by the constructed assignment. Hence, all clauses of $\phi$ are satisfied by this assignment, concluding the proof of the theorem. □

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. If we do not allow 0 costs among different colors, then the problem is inapproximable within any polynomial factor. Indeed, for any positive integer $c$ we can replace every 0 cost by $n^{-c-1}$. Then the optimum becomes at most $n^{-c}$, and the cost of a non-optimal solution is at least 1. Therefore, it is NP-hard to get an approximation ratio of $n^c$.

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.** *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.*

**Proof.** The reduction follows closely the one of Theorem 3. Given a monotone rectilinear representation of a planar monotone 3-SAT instance $\phi$, we build an instance $(H, X, \chi, r, f)$ of MinCCA as follows. The variable-vertices and the clause-vertices of $G_\phi$ are $\{x_1, \ldots, x_n\}$ and $\{C_1, \ldots, C_m\}$, respectively. Without loss of generality, we assume that the variable-vertices appear in the order $x_1, \ldots, x_n$. For every variable-vertex $x_i$ of $G_\phi$, we add to $H$ a gadget similar to the gadget used in the proof of Theorem 3, 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 $\{r, x_1^\ell\}$. Let $\mathcal{C}_i^+$ be the set of clauses that variable $i$ appears positively and let $\mathcal{C}_i^-$ be the set of clauses that variable $i$ appears negatively. For every $i \in \{1, \ldots, n\}$ and for every clause $j \in \left\{1, \ldots, |\mathcal{C}_i^+|\right\}$, we add vertices $x_{ij}^+$ and $x_{ij}^{r+}$. Likewise, for every $i \in \{1, \ldots, n\}$ and for every clause $j \in \left\{1, \ldots, |\mathcal{C}_i^-|\right\}$, we add vertices $x_{ij}^-$ and $x_{ij}^{r-}$. Moreover, for every $i \in \{1, \ldots, n-1\}$, we add a vertex $x_i'$ as well as the edges $\left\{x_i^r, x_i'\right\}$ and $\left\{x_i', x_{i+1}^l\right\}$. We proceed our construction by adding for every $i \in \{1, \ldots, n\}$ and $j \in \left\{1, \ldots, |\mathcal{C}_i^+|\right\}$ the edges $\left\{x_i^+, x_{ij}^+\right\}$, $\left\{x_i^r, x_{ij}^{r+}\right\}$, $\left\{x_{ij}^+, x_{ij}^{r+}\right\}$, $\left\{x_{ij}^+, C_j\right\}$ and for every $j \in \left\{1, \ldots, |\mathcal{C}_i^-|\right\}$ the edges $\left\{x_i^-, x_{ij}^-\right\}$, $\left\{x_i', x_{ij}^{r-}\right\}$, $\left\{x_{ij}^-, x_{ij}^{r-}\right\}$, $\left\{x_{ij}^-, C_j\right\}$. Subsequently, for every $i \in \{1, \ldots, n\}$ and $j \in \left\{1, \ldots, |\mathcal{C}_i^+| - 1\right\}$ we add the edges $\left\{x_{ij}^+, x_{i(j+1)}^+\right\}$, $\left\{x_{ij}^{r+}, x_{i(j+1)}^{r+}\right\}$, and for every $j \in \left\{1, \ldots, |\mathcal{C}_i^-| - 1\right\}$ we add the edges $\left\{x_{ij}^-, x_{i(j+1)}^-\right\}$, $\left\{x_{ij}^{r-}, x_{i(j+1)}^{r-}\right\}$. Note that the maximum degree of $H$ is indeed 4. An example of this construction can be found in Fig. 9.

We define the color palette as $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Let us now describe the edge coloring function $\chi$. For every clause vertex, we arbitrarily color its three incident edges with colors $\{4, 5, 6\}$ so that each incident edge gets a different color. For every $i \in \{1, \ldots, n\}$, 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 $\chi(\{x_i^+, x_i^-\}) = 3$.

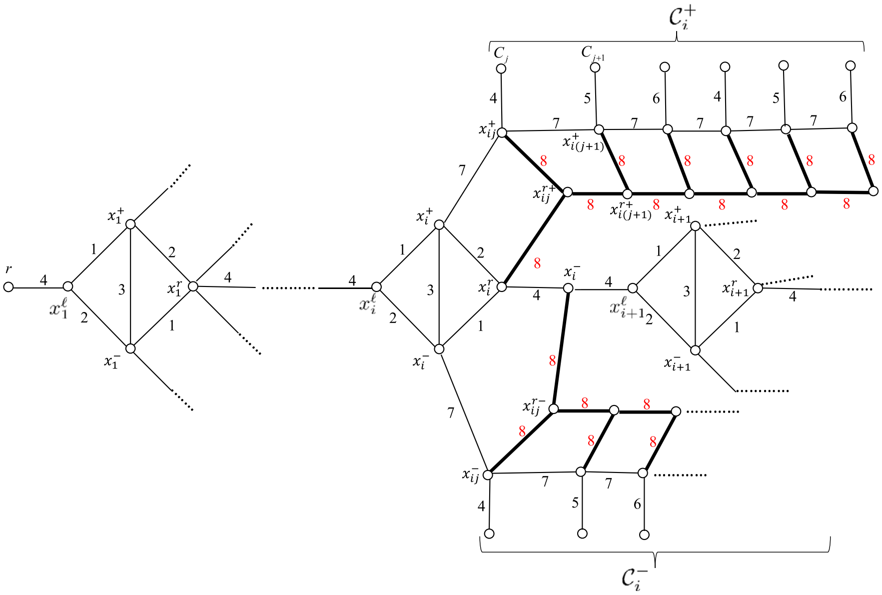


**Fig. 9.** The graph $H$ constructed in the proof of Theorem 4, together with the edge-coloring function $\chi$. Note that the maximum degree of $H$ is 4.

We set $\chi(\{r, x_1^\ell\}) = 4$ and for every $i \in \{1, \dots, n-1\}$, $\chi(\{x_i^r, x_i'\}) = \chi(x_i', x_{i+1}^l) = 4$. For every $j \in \left\{1, \dots, |\mathcal{C}_i^+| - 1\right\}$, we set $\chi(x_{ij}^+, x_{i(j+1)}^+) = 7$ and $\chi(x_{ij}^{r+}, x_{i(j+1)}^{r+}) = 8$. Likewise, for every $j \in \left\{1, \dots, |\mathcal{C}_i^-| - 1\right\}$, we set $\chi(x_{ij}^-, x_{i(j+1)}^-) = 7$ and $\chi(x_{ij}^{r-}, x_{i(j+1)}^{r-}) = 8$. For every $i \in \{1, \dots, n\}$ and $j \in \left\{1, \dots, |\mathcal{C}_i^+|\right\}$, we set $\chi(x_{ij}^+, x_{ij}^{r+}) = \chi(x_i^r, x_{ij}^{r+}) = 8$, $\chi(x_i^+, x_{ij}^+) = 7$ and for every $j \in \left\{1, \dots, |\mathcal{C}_i^-|\right\}$ we set $\chi(x_i', x_{ij}^{r-}) = \chi(x_{ij}^-, x_{ij}^{r-}) = 8$, $\chi(x_i^-, x_{ij}^-) = 7$. The function $\chi$ is also depicted in Fig. 9. Finally, we define the cost function cc as follows. We set $\mathrm{cc}(1,2) = 1$, $\mathrm{cc}(1,3) = \mathrm{cc}(2,3) = 0$, $\mathrm{cc}(1,7) = \mathrm{cc}(2,7) = 0$, and $\mathrm{cc}(3,7) = 1$. For every $i \in \{1,2,3,4,5,6,7,8\}$ we set $\mathrm{cc}(i,i) = 0$. For every $i, j \in \{4,5,6\}$ with $i \neq j$ we set $\mathrm{cc}(i,j) = 1$. For all $i \in \{1,2,3,4,5,6\}$, $\mathrm{cc}(8,i) = 0$, whereas $\mathrm{cc}(1,8) = \mathrm{cc}(2,8) = 0$ and $\mathrm{cc}(7,8) = \mathrm{cc}(8,7) = 1$. Moreover, $\mathrm{cc}(1,4) = \mathrm{cc}(2,4) = 0$ and for all $i \in \{4,5,6\}$ we set $\mathrm{cc}(8,i) = \mathrm{cc}(i,8) = 0$ and $\mathrm{cc}(7,i) = \mathrm{cc}(i,7) = 1$.

We now claim that $H$ contains an arborescence $T$ rooted at $r$ with cost 0 if and only if the formula $\phi$ is satisfiable.

Suppose first that $\phi$ is satisfiable, and we proceed to define an arborescence $T$ rooted at $r$ with cost 0. $T$ contains the edge $\{r, x_1^\ell\}$ and, for every $i \in \{1, \dots, n-1\}$, the edges $\left\{x_i^r, x_i'\right\}$ and $\left\{x_i', x_{i+1}^\ell\right\}$. For every $i \in \{1, \dots, n\}$, if variable $x_i$ is set to 1 in the satisfying assignment of $\phi$, we add to $T$ the edges $\{x_i^\ell, x_i^+\}$, $\{x_i^+, x_i^-\}$, and $\{x_i^-, x_i^r\}$. Otherwise, if variable $x_i$ is set to 0 in the satisfying assignment of $\phi$, we add to $T$ the edges $\{x_i^\ell, x_i^-\}$, $\{x_i^-, x_i^+\}$, and $\{x_i^+, x_i^r\}$. For every $i \in \{1, \dots, n\}$, if variable $x_i$ is set to 1 in the satisfying assignment of $\phi$, then for every $j \in \left\{1, \dots, |C_i^+|\right\}$ we add to $T$ the edges $\left\{x_i^r, x_{ij}^{r+}\right\}$, $\left\{x_{ij}^{r+}, x_{ij}^+\right\}$, $\left\{x_{ij}^+, C_j\right\}$ (note that for each clause $C_j$ we add only one such edge), for every $j \in \left\{1, \dots, |C_i^+| - 1\right\}$ we add to $T$ the edges $\left\{x_{ij}^{r+}, x_{i(j+1)}^{r+}\right\}$, for every $j \in \left\{1, \dots, |C_i^-|\right\}$ we add to $T$ the edges $\left\{x_i^-, x_{ij}^-\right\}$, $\left\{x_i', x_{ij}^{r-}\right\}$, and for every $j \in \left\{1, \dots, |C_i^-| - 1\right\}$ we add to $T$ the edges $\left\{x_{ij}^{r-}, x_{i(j+1)}^{r-}\right\}$, $\left\{x_{ij}^-, x_{i(j+1)}^-\right\}$. For every $i \in \{1, \dots, n\}$, if variable $x_i$ is set to 0 in the satisfying assignment of $\phi$, then for every $j \in \left\{1, \dots, |C_i^+|\right\}$ we add to $T$ the edges $\left\{x_i^r, x_{ij}^{r+}\right\}$, $\left\{x_i^+, x_{ij}^+\right\}$, for every $j \in \left\{1, \dots, |C_i^+| - 1\right\}$ we add to $T$ the edges $\left\{x_{ij}^{r+}, x_{i(j+1)}^{r+}\right\}$, $\left\{x_{ij}^+, x_{i(j+1)}^+\right\}$, for every $j \in \left\{1, \dots, |C_i^-|\right\}$ we add to $T$ the edges $\left\{x_i', x_{ij}^{r-}\right\}$, $\left\{x_{ij}^{r-}, x_{ij}^-\right\}$, $\left\{x_{ij}^-, C_j\right\}$ (note that for each clause $C_j$ we add only one such edge), and for every $j \in \left\{1, \dots, |C_i^-| - 1\right\}$ we add to $T$ the edges $\left\{x_{ij}^{r-}, x_{i(j+1)}^{r-}\right\}$.

Conversely, suppose now that $H$ contains an arborescence $T$ rooted at $r$ with cost at most 0 and let us define a satisfying assignment of $\phi$. Since for every $i, j \in \{4,5,6\}$ with $i \neq j$ we have that $\mathrm{cc}(i,j) = 1$, for every clause-vertex $C_j$ exactly one of its incident edges belongs to $T$. Since $\mathrm{cc}(i,8) = 0$ and $\mathrm{cc}(i,7) = 1$ for all $i \in \{4,5,6\}$, we have that if $\left\{x_{ij}^+, C_j\right\}$ belongs to $T$, then $\left\{x_{ij}^+, x_{ij}^{r+}\right\}$, $\left\{x_{ij}^{r+}, x_i^r\right\}$, and all other edges with color 8 in the gadget corresponding to $C_i^+$ belong to $T$. Due to the same reasons, if $\left\{x_{ij}^-, C_j\right\}$ belongs to $T$, then $\left\{x_{ij}^-, x_{ij}^{r-}\right\}$, $\left\{x_{ij}^{r-}, x_i'\right\}$, and all edges with color 8 in the gadget corresponding to $C_i^-$ belong to $T$. From the structure of $H$ and from the fact that $\mathrm{cc}(1,2) = 1$, it follows that in order for the tree $T$ to span all vertices of $H$, for every $i \in \{1, \dots, n\}$ either the three edges $\{x_i^\ell, x_i^+\}$, $\{x_i^+, x_i^-\}$, $\{x_i^-, x_i^r\}$ or the three edges $\{x_i^\ell, x_i^-\}$, $\{x_i^-, x_i^+\}$, $\{x_i^+, x_i^r\}$ belong to $T$. In the former case, we set variable $x_i$ to 1, and in the latter case we set variable $x_i$ to 0. Since for every $i \in \{4,5,6\}$ we have that $\mathrm{cc}(7,i) = 1$, it follows that for every clause-vertex $C_j$, its incident edge that belongs to $T$ joins $C_j$ to a literal that is set to 1 by the constructed assignment. Hence, all clauses of $\phi$ are satisfied by this assignment, concluding the proof of the theorem. $\square$

From Theorem 4, it is not difficult to prove that the MinCCA problem is NP-hard even on grids. The main idea is that any planar graph $G$ with maximum degree at most 4 is contained in a grid as a topological minor. From this observation, one can play with the costs in such a way that solving the problem in $G$ is equivalent to solving it in the grid on which $G$ is embedded.

## 6. Conclusions and further research

In this article we proved several hardness results for the MinCCA problem. Our main result, which answers an open question from Gözüpek et al. [14], is that the problem is W[1]-hard parameterized by the vertex cover number (hence, by treewidth as well) on general graphs. We also 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.

As a partial result in this direction, in an extended version of this article, permanently available at [arxiv:1605.00532], we provide an FPT algorithm for the MinCCA problem parameterized by a *restricted* version of tree-cutwidth. Without entering into technical details here, in this variation of tree-cutwidth we further impose a *limited* dependency among the *thin* children of each node $t$ of the tree-cut decomposition. According to Ganian et al. [11], a non-root node $t$ of $T$ is *thin* if $\mathrm{adh}(t) \leq 2$. While this parameter is somehow artificial, the algorithm shows that bounded tree-cutwidth is "almost" enough to provide an FPT algorithm for MinCCA, in the sense that the potential source of hardness, if any, comes from the structure of thin nodes. It is worth mentioning that this algorithm can be seen as a generalization of the FPT algorithm of Gözüpek et al. [14] with parameter $\mathbf{tw} + \Delta$. Indeed, if the maximum degree is also considered as a parameter, then the number of thin children in a tree-cut decomposition is bounded at any node, and we can solve the problem in time FPT using our algorithm.

Our hardness results imply that the problem is para-NP-hard when parameterized by the number of distinct cost values. Another interesting research direction is to consider parameters related to the cost function, such as the value of the largest cost, the ratio of the largest cost to the smallest cost, the ratio the largest cost to the smallest cost among different colors, costs obeying the triangle inequality, etc.

On the other hand, we proved that the MinCCA problem is NP-hard on planar graphs; however, we do not know whether it is W[1]-hard parameterized by treewidth on planar graphs.

It would be natural to consider other structural parameters as well as other width parameters. Note that the MinCCA problem is W[1]-hard parameterized by any parameter that can by bounded as a function of the vertex cover number, such as treedepth or branchwidth. A well-known parameter for which the problem is para-NP-hard is cliquewidth. Indeed, given an arbitrary graph $G$ on $n$ vertices as instance of MinCCA, let $G'$ be the graph obtained from a clique on $n$ vertices, corresponding to the vertices of $G$, by adding a new pendant vertex to each vertex of the clique. The transition costs among edges of $G'$ that were already in $G$ remain unchanged, the ones involving pendant edges are set to zero, and the other ones to infinity. It is easy to verify that solving MinCCA in $G$ and in $G'$ is equivalent, and that $G'$ has cliquewidth 2.

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 [5,8,10,20].

## Acknowledgement

We would like to thank the anonymous referees of this article and its conference version for helpful comments that improved the presentation of the manuscript.

## References

[1] S. Arkoulis, E. Anifantis, V. Karyotis, S. Papavassiliou, N. Mitrou, On the optimal, fair and channel-aware cognitive radio network reconfiguration, Comput. Netw. 57 (8) (2013) 1739–1757.
[2] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.
[3] M. de Berg, A. Khosravi, Optimal binary space partitions for segments in the plane, Internat. J. Comput. Geom. Appl. 22 (3) (2012) 187–206.
[4] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013.
[5] M.R. Çelenlioğlu, D. Gözüpek, H.A. Mantar, A survey on the energy efficiency of vertical handover mechanisms, in: Proc. of the International Conference on Wireless and Mobile Networks, WiMoN, 2013.
[6] M.R. Fellows, D. Hermelin, F. Rosamond, S. Vialette, On the parameterized complexity of multiple-interval graph problems, Theoret. Comput. Sci. 410 (1) (2009) 53–61.
[7] J. Flum, M. Grohe, Parameterized Complexity Theory, Texts in Theoretical Computer Science, Springer, 2006.
[8] G. Galbiati, The complexity of a minimum reload cost diameter problem, Discrete Appl. Math. 156 (18) (2008) 3494–3497.
[9] G. Galbiati, S. Gualandi, F. Maffioli, On minimum changeover cost arborescences, in: Proc. of the 10th International Symposium on Experimental Algorithms, SEA, in: LNCS, vol. 6630, 2011, pp. 112–123.
[10] G. Galbiati, S. Gualandi, F. Maffioli, On minimum reload cost cycle cover, Discrete Appl. Math. 164 (2014) 112–120.

[11] R. Ganian, E.J. Kim, S. Szeider, Algorithmic applications of tree-cut width, in: Proc. of the 40th International Symposium on Mathematical Foundations of Computer Science, MFCS, in: LNCS, vol. 9235, 2015, pp. 348–360.
[12] L. Gourvès, A. Lyra, C. Martinhon, J. Monnot, The minimum reload s-t path, trail and walk problems, Discrete Appl. Math. 158 (13) (2010) 1404–1417.
[13] D. Gözüpek, S. Buhari, F. Alagoz, A spectrum switching delay-aware scheduling algorithm for centralized cognitive radio networks, IEEE Trans. Mob. Comput. 12 (7) (2013) 1270–1280.
[14] D. Gözüpek, H. Shachnai, M. Shalom, S. Zaks, Constructing minimum changeover cost arborescenses in bounded treewidth graphs, Theoret. Comput. Sci. 621 (2016) 22–36.
[15] D. Gözüpek, M. Shalom, A. Voloshin, S. Zaks, On the complexity of constructing minimum changeover cost arborescences, Theoret. Comput. Sci. 540 (2014) 40–52.
[16] F.V. Jensen, Bayesian Networks and Decision Graphs, Springer, 2001.
[17] E.J. Kim, S.-I. Oum, C. Paul, I. Sau, D.M. Thilikos, An FPT 2-approximation for tree-cut decomposition, Algorithmica (2017), http://dx.doi.org/10.1007/s00453-016-0245-5, in press.
[18] S.J. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. R. Stat. Soc., B 50 (1988) 157–224.
[19] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, vol. 31, Oxford University Press, 2006.
[20] H.-C. Wirth, J. Steffan, Reload cost problems: minimum diameter spanning tree, Discrete Appl. Math. 113 (1) (2001) 73–85.
[21] P. Wollan, The structure of graphs not admitting a fixed immersion, J. Combin. Theory Ser. B 110 (2015) 47–66.