

Minimum Reload Cost Graph Factors

Julien Baste, Didem Gözüpek, Mordechai Shalom, Dimitrios M. Thilikos

▶ **To cite this version:**

Julien Baste, Didem Gözüpek, Mordechai Shalom, Dimitrios M. Thilikos. Minimum Reload Cost Graph Factors. Theory of Computing Systems, 2021, 65 (5), pp.815-838. 10.1007/s00224-020-10012-x . hal-03327294

HAL Id: hal-03327294
https://hal.science/hal-03327294

Submitted on 22 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Minimum Reload Cost Graph Factors Graphs[1]

Julien Baste[2] Didem Gözüpek[3] Mordechai Shalom[4] Dimitrios M. Thilikos[5,6]

Abstract

The concept of *Reload cost* in a graph refers to the cost that occurs while traversing a vertex via two of its incident edges. This cost is uniquely determined by the colors of the two edges. This concept has various applications in transportation networks, communication networks, and energy distribution networks. Various problems using this model are defined and studied in the literature. The problem of finding a spanning tree whose diameter with respect to the reload costs is smallest possible, the problems of finding a path, trail or walk with minimum total reload cost between two given vertices, problems about finding a proper edge coloring of a graph such that the total reload cost is minimized, the problem of finding a spanning tree such that the sum of the reload costs of all paths between all pairs of vertices is minimized, and the problem of finding a set of cycles of minimum reload cost, that cover all the vertices of a graph, are examples of such problems. In this work we focus on the last problem. Noting that a cycle cover of a graph is a 2-factor of it, we generalize the problem the the problem of finding an r-factor of minimum reload cost of an edge colored graph. We prove several NP-hardness results for special cases of the problem. Namely, bounded degree graphs, planar graphs, bounded total cost, and bounded number of distinct costs. For the special case of $r = 2$, our results imply an improved NP-hardness result. On the positive side, we present a polynomial-time solvable special case which provides a tight boundary between the polynomial and hard cases in terms of r and the and the maximum degree of the graph. We then investigate the parameterized complexity of the problem, prove W[1]-hardness results and present an FPT-algorithm.

keywords: Parameterized Complexity, Graph Factors, Reload Costs

1 Introduction

Edge-colored graphs can be used to model optimization problems in diverse fields such as bioinformatics, communication networks, and transportation networks. *Reload cost* in an edge-colored graph refers to the cost that occurs while traversing a vertex via two of its incident edges. This cost is uniquely determined by the colors of the two edges.

[1]Emails: julien.baste@lip6.fr, didem.gozupek@gtu.edu.tr, cmshalom@telhai.ac.il, sedthilk@thilikos.info.
[2]Sorbonne Université, Laboratoire d'Informatique de Paris 6, LIP6, Paris, France
[3]Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey.
[4]TelHai College, Upper Galilee, 12210, Israel.
[5]AlGCo project-team, LIRMM, CNRS, Université de Montpellier, Montpellier, France.
[6]Supported by projects DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010) and by the Research Council of Norway and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA 2019.

The reload cost concept has various applications in transportation networks, communication networks, and energy distribution networks. For instance, a multi-modal cargo transportation network involves different means of transportation, where the (un)loading of cargo at transfer points is costly [9]. In energy distribution networks, transferring energy between different carriers cause energy losses and reload cost concept can be used to model this situation [9]. In communication networks, routing often requires switching between different technologies such as cable and fiber, where data conversion incurs high costs. Switching between different service providers in a communication network also causes switching costs [9]. Recently, *dynamic spectrum access networks*, a.k.a. cognitive radio networks, received a lot of attention in the communication networks research community. Unlike other wireless networks, cognitive radio networks are envisioned to operate in a wide range of spectrum; therefore, frequency switching has adverse effects in delay and energy consumption [2,5,15]. This frequency switching cost depends on the frequency separation distance; hence, it corresponds to reload costs.

The reload cost concept was first introduced by Wirth and Steffan [27] who focused on the problem of finding a spanning tree whose diameter with respect to the reload costs is smallest possible. Other works also focused on numerous optimization problems regarding reload costs: the problems of finding a path, trail or walk with minimum total reload cost between two given vertices [13], numerous path, tour, and flow problems [1], the minimum changeover cost arborescence problem [10,14,16,18], problems about finding a proper edge coloring of a graph such that the total reload cost is minimized [17], and the problem of finding a spanning tree such that the sum of the reload costs of all paths between all pairs of vertices is minimized [12].

An r-factor of a graph is an r-regular spanning subgraph. A 2-factor is also called a *cycle cover* and has many applications in areas such as computer graphics and computational geometry [21], for instance for fast rendering of 3D scenes. In an edge-weighted graph, the problem of finding a cycle cover with minimum cost can be solved in polynomial-time [25]. Its reload cost counterpart was studied by Galbiati et.al. in [11]. In particular, they proved that the minimum reload cost cycle cover problem is NP-hard even when the number of colors is 2, the reload costs are symmetric and satisfy the triangle inequality. In this work, we build on this work by studying the minimum reload cost r-factor problem, which is a more generalized version of the minimum reload cost cycle cover problem. In particular, we prove several NP-hardness results for the special cases of this problem. Namely, bounded degree graphs, planar graphs, bounded total cost, and bounded number of distinct costs. For the special case of $r = 2$, we prove an NP-hardness result stronger than the one in [11]. On the positive side, we present a polynomial-time solvable special case. We then investigate the parameterized complexity of this problem, prove W[1]-hardness results and present a fixed parameter tractable algorithm.

2 Preliminaries

Sets, vectors: Given a non-negative integer n, we denote by $\mathbb{N}_{\geq n}$ the set of all integers x such that $x \geq n$. If $n_1, n_2 \in \mathbb{N}_{\geq 0}$, we denote by $[n_1, n_2]$ the set of integers x such that $n_1 \leq x \leq n_2$. We also use $[n]$ instead of $[1, n]$. Given a finite set X and an integer $s \in \mathbb{N}_{\geq 0}$, we denote by $\binom{X}{s}$ the set of all subsets of A with exactly s elements. For a set X and an element x we use $X + x$ and $X - x$ as shorthands for $X \cup \{x\}$ and $X \setminus \{x\}$, respectively. For two vectors $\mathbf{u} = (u_1, \ldots, u_d)$ and $\mathbf{v} = (v_1, \ldots, v_d)$ over the reals, we write $\mathbf{u} \leq \mathbf{v}$ if $u_i \leq v_i$ for every $i \in [d]$.

Graphs: All graphs we consider in this paper are undirected, finite, and without self loops or multiple edges. Given a graph G, we denote by $V(G)$ the set of vertices of G and by $E(G)$ the set of edges of G. We say that a vertex $v \in V(G)$ and an edge $e \in E(G)$ are *incident* if $v \in e$, that is, v is an endpoint of e. Given a vertex $v \in V(G)$, we denote by $E_G(v)$ the set of edges of G that are incident to v. The *degree* of v in G, denoted by $\mathsf{deg}_G(v)$ is $|E_G(v)|$. We also define the *maximum degree* of G as $\Delta(G) = \max\{\mathsf{deg}_G(v) \mid v \in V(G)\}$, the *minimum degree* of G as $\delta(G) = \min\{\mathsf{deg}_G(v) \mid v \in V(G)\}$, and the average degree of G as $\overline{\mathsf{deg}}(G)$. For a subset X of $V(G)$, we denote by $G[X]$ the subgraph of G induced by X. A graph is *r-regular* if all its vertices have degree r.

We say that a graph H is a *factor* of a graph G when $V(H) = V(G)$ and $E(H) \subseteq E(G)$. An r-regular factor of G is termed an *r-factor* of G.

Parameterized complexity: We refer the reader to [6,8] for basic background on parameterized complexity, and we recall here only some basic definitions. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$. For an instance $I = (x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*.

A parameterized problem is *fixed-parameter tractable* (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ (called an FPT *algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^c$.

A parameterized problem is in XP if there exists an algorithm $\mathcal{A}$ and two computable functions f and g such that given an instance $I = (x, k)$, $\mathcal{A}$ (called an XP *algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^{g(k)}$.

A parameterized problem with instances of the form $I = (x, k)$ is para-NP-*hard* if it is NP-hard for some fixed *constant* value of the parameter k. Note that, unless P = NP, a para-NP-hard problem cannot be in XP, hence it cannot be FPT either.

Within parameterized problems, the class W[1] may be seen as the parameterized equivalent to the class NP of classical optimization problems. Without entering into details (see [6,8] for the formal definitions), a parameterized problem being W[1]-hard can be seen as a strong evidence that this problem is *not* FPT. To transfer W[1]-hardness from one problem to another, one uses an FPT reduction, which given an input $I = (x, k)$ of the source problem, computes in time $f(k) \cdot |I|^c$, for some computable function f and a function c, an equivalent instance $I' = (x', k')$ of the target problem such that k' is bounded by a function depending only on k. The following problem is a W[1]-hard problem that we will use in our reductions.

Multicolored Clique
Input: A graph G, an integer k, a coloring function $\chi : V(G) \to [k]$.
Parameter: k.
Question: Does G contain a clique on k vertices with one vertex from each color class?

Multicolored 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 [22]. Clearly, we can also assume that every color class is an independent set since the problem is indifferent to edges within the same color class.

Tree decompositions: A *tree decomposition* of a graph $G = (V, E)$ is a pair $\mathcal{D} = (T, \mathcal{X})$, where T is a tree and $\mathcal{X} = \{X_t \mid t \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

- $\bigcup_{t \in V(T)} X_t = V$,

- for every edge $uv \in E$, there is a $t \in V(T)$ such that $\{u, v\} \subseteq X_t$, and
- for every $\{x, y, z\} \subseteq V(T)$ such that z lies on the unique path between x and y in T, $X_x \cap X_y \subseteq X_z$.

We call the vertices of T *nodes* of $\mathcal{D}$ and the sets in $\mathcal{X}$ *bags* of $\mathcal{D}$. The width of $\mathcal{D}$ is $\max_{t \in V(T)} |X_t| - 1$. The *treewidth* of G, denoted by $\mathsf{tw}(G)$, is the smallest integer w such that there exists a tree decomposition of G of width w. A tree decomposition in which the tree T is restricted to be a path is called a *path decomposition*. The *pathwidth* of G, denoted by $\mathsf{pw}(G)$, is the smallest integer w such that there exists a path decomposition of G of width w.

A tree decomposition is *rooted* if we distinguish in T some specific vertex r, and consider T as a rooted (on r) tree. We denote such a tree decomposition by a triple $\mathcal{D} = (T, \mathcal{X}, r)$.

Nice tree decompositions: Let $\mathcal{D} = (T, \mathcal{X}, r)$ be a rooted tree decomposition of G, and $\mathcal{G} = \{G_t \mid t \in V(T)\}$ be a collection of subgraphs of G. We say that the ordered pair $(\mathcal{D}, \mathcal{G})$ is *nice* if the following conditions hold:

- $X_r = \emptyset$ and $G_r = G$,
- every node of $\mathcal{D}$ has at most two children in T,
- for each leaf $t \in V(T)$, $X_t = \emptyset$ and $G_t = (\emptyset, \emptyset)$. Such a node t is called a *leaf node*,
- if $t \in V(T)$ has exactly two children t' and t'', then $X_t = X_{t'} = X_{t''}$, $G_t = G_{t'} \cup G_{t''}$, and $E(G_{t'}) \cap E(G_{t''}) = \emptyset$. The node t is called a *join node*.
- if $t \in V(T)$ has exactly one child t', then exactly one of the following holds.
  - $X_t = X_{t'} + v$ for some $v \notin X_{t'}$ and $G_t = (V(G_{t'}) + v, E(G_{t'}))$. The node t is called *vertex-introduce node* and the vertex v is the *introduced vertex* of X_t.
  - $X_t = X_{t'}$ and $G_t = (G_{t'}, E(G_{t'}) + e)$ where e is an edge of G with endpoints in X_t. The node t is called *edge-introduce node* and the edge e is the *introduced edge* of X_t.
  - $X_t = X_{t'} - v$ for some $v \in X_{t'}$ and $G_t = G_{t'}$. The node t is called *forget node* and v is the *forget vertex* of X_t.

The notion of a nice pair defined above is essentially the same as the one of nice tree decomposition in [7] (which in turn is an enhancement of the original one, introduced in [20]). As already argued in [7, 20], given a tree decomposition, it is possible to transform it in polynomial time to a tree decomposition $\mathcal{D}$ of the same width and construct a collection $\mathcal{G}$ such that $(\mathcal{D}, \mathcal{G})$ is nice.

Reload cost model: For reload costs, we follow the notation and terminology defined by [27]. We consider an edge-colored graph G where edge colors are taken from a finite set X and the coloring function is $\chi : E(G) \to X$. The reload costs are given by a function $c : X^2 \to \mathbb{N}_{\geq 0}$ where $c(x_1, x_2) = c(x_2, x_1)$ for each $(x_1, x_2) \in X^2$. The cost of *traversing* two incident edges e_1, e_2 of G is $\mathsf{tc}(e_1, e_2) = c(\chi(e_1), \chi(e_2))$. Given a subgraph H of G and a vertex $v \in V(H)$, we define the *reload cost* of v in H as $\mathsf{rc}_{\chi,c}(H, v) = \sum_{\{e_1,e_2\} \in \binom{E_H(v)}{2}} \mathsf{tc}(e_1, e_2)$ and the *reload cost* of H as $\mathsf{rc}_{\chi,c}(H) = \sum_{v \in V(H)} \mathsf{rc}_{\chi,c}(H, v)$. When χ and c are clear from the context, we write $\mathsf{rc}(v)$ and $\mathsf{rc}(H)$ instead.

Problem statement: The problem we study in this paper can be formally defined as follows for every $r \in \mathbb{N}_{\geq 2}$:

Minimum Reload Cost r-Factor (r-MRCF)
Input: A graph G, an edge-coloring χ, a reload cost function c, and a non-negative integer k.
Output: Is there an r-factor H of G with reload cost at most k, i.e., $\mathsf{rc}(H) \leq k$?

Given an instance (G, χ, c, k) of r-MRCF we consider the following parameters:

- the maximum degree $\Delta(G)$ of G,
- the treewidth $\mathsf{tw}(G)$ of G,
- the number of colors $q = |X|$,
- the number of distinct costs: $d = |\{c(x_1, x_2) \mid (x_1, x_2) \in X^2\}|$,
- the minimum traversal cost $c_{min} = \min\{c(x_1, x_2) \mid (x_1, x_2) \in X^2\}$, and
- the total cost k.

Clearly, we can assume that $\Delta(G) \geq r+1$, and also $\delta(G) \geq r$ since otherwise the instance is trivial. Let $k_{\min} = c_{min} \cdot |V(G)| \cdot \binom{r}{2}$. Note that the reload cost of every r-factor is at least $k_{\min}$. Therefore, we can also assume that $k \geq k_{\min}$.

A summary of the results regarding the classical and parameterized complexity of the r-MRCF problem is shown in Table 1 and Table 2, respectively.

G	$\Delta(G)$	d	k	q	
	$\leq r+2$	2	k_{min}	$\min\{r, 3\}$	NP-hard (Theorem 1)
	$r+1$				Polynomial (Theorem 2)
Planar	$\leq r+4$	2	k_{min}	7	NP-hard (Theorem 3)

Table 1: Summary of classical complexity results for the r-MRCF problem.

Parameter	d	k	average degree	
$\mathsf{pw}(G)$	2	k_{min}	$< r + \begin{cases} 4 & \text{if } r = 2 \\ 4/3 & \text{otherwise.} \end{cases}$	W[1]-hard (Theorem 4)
$\mathsf{tw}(G) + \min\{q, \Delta(G)\}$				FPT(Theorem 5)

Table 2: Summary of parameterized complexity results for the r-MRCF problem.

3 Classical Complexity of r-MRCF

The following construct will be used in our reductions. A *diamond* is a graph on four vertices and five edges, that is obtained by adding one chord to a cycle on four vertices. Clearly, a diamond contains two vertices of degree two and two vertices of degree three. The degree two vertices are

termed as the *tip*s of the diamond. A *joker* is a monochromatic diamond, that is, a diamond whose edges have the same color. In our reductions, every a joker J will have exactly one vertex adjacent to other vertices of the graph. This vertex will always be a tip of J, and we will term it as the *connecting tip* of J. Given a joker J and a 2-factor F, it is easy to see that exactly one of the following happens:

- $F \cap E(J)$ is the 4-cycle of J, and $F \setminus E(J)$ does not contain any edges incident to J.
- $F \cap E(J)$ is a triangle of J, $F \setminus E(J)$ contains exactly two edges incident to J both of which are incident its connecting tip.

Furthemore, since our cost functions satisfy $c(\lambda, \lambda) = 0$ for every color λ, and $F \cap E(J)$ is always a cycle, the joker edges do not affect the cost of F. When we describe a 2-factor F, these properties allow us to leave the edges $F \cap E(J)$ unspecified since they are implied by the edges of $F \setminus E(J)$. Such a partial description is valid if and only if the connecting tip have degree zero or two. Finally, a 5-*joker* is a cycle on five vertices with an added chord. This graph has one triangle with one degree 2 vertex that we will refer to as *the tip* of the 5-joker. Note that a 5-joker has all the properties of a joker.

Another construct that we use in our reductions is a graph Q_ℓ that is obtained from the clique on $\ell + 1$ vertices by subdividing $\ell - 2$ arbitrary edges twice (into three edges) and removing the middle edge of each one. Clearly, Q_ℓ contains $\ell + 1$ vertices of degree ℓ and $2(\ell - 2)$ vertices of degree one. In total G_ℓ has $3\ell - 3$ vertices.

Galbiati et.al. proved in [11] that 2-MRCF, a.k.a. the minimum reload cost cycle cover problem, is NP-hard even when the number of colors is 2, the reload costs are symmetric and satisfy the triangle inequality. In the following, we obtain a hardness result for r-MRCF, which in particular implies a stronger hardness result for the special case of $r = 2$.

Theorem 1. r-MRCF *is* NP*-hard for every* $r \geq 2$ *even when* $\Delta(G) \leq r + 2$, $d = 2$, $k = k_{\min} = 0$ *and* $q = \min\{r, 3\}$.

Proof. We start by proving the claim for $r = 2$, that is, we prove that 2-MRCF is NP-hard when $\Delta(G) \leq 4$, $d = q = 2$, $k = k_{\min}$. The proof is by reduction from the MONOTONE 1 IN 3-SAT problem which is known to be NP-hard [24]. An instance of MONOTONE 1 IN 3-SAT is an expression ϕ that consists of n boolean variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$. Every clause c_j consists of three literals $\ell_{j,1}, \ell_{j,2}, \ell_{j,3}$, each of which is either an occurrence of some variable x_i or its negation $\bar{x}_i$. A clause is satisfied by a truth assignment if exactly one of its literals is satisfied by it, and ϕ is satisfied if all its clauses are satisfied. MONOTONE 1 IN 3-SAT is the problem of determining whether a given expression ϕ has a satisfying assignment.

Given an instance ϕ as described above, we construct an instance (G, χ, c, k) of 2-MRCF. The set of colors is $X = \{red, blue\}$, thus $q = 2$. The traversal cost between two identical colors is zero and the traversal cost between red and blue is 1, thus $d = 2$. Furthermore, $k = 0$. We have $c_{min} = 0$, which implies that $k_{\min} = 0 = k$. To complete the reduction, we construct a graph G with $\Delta(G) = 4$ whose edges are colored with colors from X.

The graph G is the union of n vertex disjoint variable gadgets with vertex sets $X_1, \ldots, X_n$, m vertex disjoint clause gadgets with vertex sets $C_1, \ldots, C_m$, and $3m + 4n$ vertex disjoint jokers, that is, four jokers $J_i^{(u)}$, $J_i^{(v)}$, $J_i^{(2)}$, $J_i^{(4)}$ for every variable x_i and a joker $J_{i,k'}$ for every occurrence of it. In our construction, every edge will join two vertices of the same gadget. However, we will allow

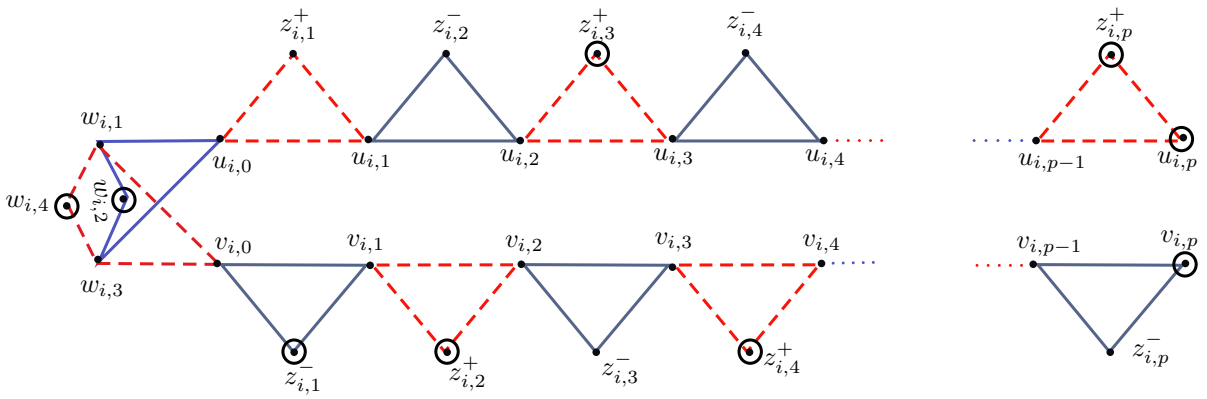


Figure 1: The variable gadget in Theorem 1 corresponding to a variable x_i having p occurrences in ϕ. The solid edges are blue, the dashed edges are red. Every vertex surrounded with a circle is the connecting tip of a joker. For every $k' \in [p]$ exactly one of $z^+_{i,k'}, z^-_{i,k'}$ is a connecting tip. The gadget contains $p+4$ connecting tips.

a vertex to appear in two (but not more) of the above gadgets. Every gadget will have maximum degree 4, where vertices shared with another gadget have degree only 2 inside the gadget. This will ensure that $\Delta(G) = 4$. We now proceed with the description of the individual gadgets.

Let x_i be a variable that has p occurrences in ϕ. Consult Figure 1 for the description of the corresponding variable gadget. We set $X_i = U_i \cup V_i \cup W_i \cup Z^+_i \cup Z^-_i$ where $U_i = \{u_{i,0}, u_{i,1}, \ldots, u_{i,p}\}$, $V_i = \{v_{i,0}, v_{i,1}, \ldots, v_{i,p}\}$, $W_i = \{w_{i,1}, w_{i,2}, w_{i,3}, w_{i,4}\}$ $Z^D_i = \{z^D_{i,1}, \ldots, z^D_{i,p}\}$ for $D \in \{+,-\}$. The edge set induced by X_i is the disjoint union of p red triangles, p blue triangles, a blue cycle on the four vertices $u_{i,0}, w_{i,1}, w_{i,2}, w_{i,3}$ and a red cycle on the four vertices $v_{i,0}, w_{i,1}, w_{i,4}, w_{i,3}$ as depicted in Figure 1. Note that every vertex of Z^+_i (resp. Z^-_i) has degree two, and both its incident edges are red (resp. blue).

For every clause c_j, the gadget C_i has six vertices $a_j, b_j, d_j, s_{j,1}, s_{j,2}, s_{j,3}$ connected with blue edges as depicted in Figure 2.

We identify each of the vertices $w_{i,2}$, $w_{i,4}$, $u_{i,k}$ and $v_{i,k}$ with a tips of the jokers $J^{(2)}_i$, $J^{(4)}_i$, $J^{(u)}_i$, and $J^{(v)}_i$, respectively. Finally, for every literal $\ell_{j,k}$ that is the k'-th occurrence of some variable x_i we identify the vertex $s_{j,k}$ with the vertex $z^+_{i,k'}$ (resp. $z^-_{i,k'}$) if the occurrence is positive (resp. negative) and we identify the vertex $z^-_{i,k'}$ (resp. $z^+_{i,k'}$) with a tip of the joker that corresponds to this literal.

To complete this part of the proof, it remains to show that ϕ has a satisfying assignment if and only if G contains a 2-factor H with zero reload cost. We suppose that ϕ has a satisfying assignment and provide a partial description of the 2-factor H incrementally. Recall that a partial description is valid if and only if every vertex has degree two, except possibly some connecting tips that are isolated. Let x_i be a variable assigned true by the assignment. $H[X_i]$ consists of all the blue edges of $G[X_i]$. Clearly, the traversal cost between these edges is zero. It is also easy to see that all the vertices of $H[X_i]$ have degree two except the vertices Z^+_i, $w_{i,4}$ and one of $u_{i,p}, v_{i,p}$ (depending on the parity of p) all of which are isolated. Note that $u_{i,p}, v_{i,p}$, $w_{i,4}$ and all the vertices of Z^+_i corresponding to negative occurrences of x_i are connecting tips of some joker. We remain with the vertices of Z^+_i corresponding to positive occurrences of x_i that are isolated but should

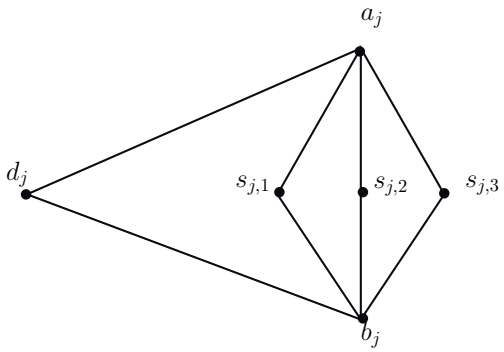


Figure 2: The clause gadget corresponding to the clause c_j.

have degree two in order to make (the partial description of) H valid. Proceeding in the same way for the variables assigned false, we get a subgraph H in which all the vertices of the variable gadgets which are not connecting tips have degree two, except the vertices of Z_i^+ corresponding to positive occurrences of variables assigned true and the vertices of Z_i^- corresponding to negative occurrences of variables assigned false. In other words, these are exactly the vertices of $Z_i^+ \cup Z_i^-$ that correspond to literals satisfied by the assignment. By adding to H the cycle $a_j d_j b_j s_{j,p}$ for every clause c_j (where $\ell_{j,p}$ is its unique literal satisfied by the assignment), we get a 2-factor H of zero cost.

Conversely, suppose that G contains a 2-factor H of zero reload cost. Consider the vertices U_i of the gadget X_i. Since every such vertex has two incident blue edges and two incident red edges, H contains either the two blue edges or the two red edges. We conclude that the vertices of $U_i - u_{i,0}$ are covered either by all the red triangles incident to them or all the blue ones. By symmetry, the same holds for the vertices of $V_i - v_{i,0}$. However, we note that the choice of the color (blue or red) must be identical in both U_i and V_i since otherwise a positive traversal cost is incurred at each of the vertices $w_{i,1}, w_{i,3}$. Therefore, $H[X_i]$ is the subgraph of $G[X_i]$ that consists of all the blue (or all the red) edges. We assign true to x_i if $H[X_i]$ contains the blue edges and false otherwise. It remains to show that this assignment satisfies ϕ. Consider a clause gadget C_j. H must contain the two edges $d_j a_j$ and $d_j b_j$. Furthermore, since every vertex $s_{j,k}$ is identified with a z vertex, H contains either both edges of the clause gadget incident to $s_{j,k}$ or none of them. Since a_j has degree 2 in H, exactly one of these three vertices, say $s_{j,1}$, has both incident edges of the clause gadget in H. Then, the two edges incident to $s_{j,1}$ in the corresponding variable gadget are not in H. This implies that either $\ell_{j,1}$ is a positive occurrence of a positive variable, or a negative occurrence of a negative variable. In other words, $\ell_{j,1}$ is satisfied. The vertices $s_{j,2}$ and $s_{j,3}$ must have in H their incident (red or blue) edges in their corresponding variable gadgets. Then, each one of them is either a positive occurrence of a negative variable, or a negative occurrence of a positive variable. In other words, their corresponding literals are not satisfied. We conclude that C_j is satisfied, and thus ϕ is satisfied.

We now prove the claim for any $r \geq 3$. We need to show that r-MRCF is NP-hard even when $t = 2$, $q = 3$, $k = k_{\min} = 0$ and $\Delta \leq r + 2$. Given an instance (G, χ, c, k) of 2-MRCF with $\Delta \leq 4$, $t = q = 2$ and $k = k_{\min} = 0$, we construct an instance (G', χ', c', k') of r-MRCF as follows. By eventually adding a monochromatic cycle of size 3, we may assume that G has an even number of

vertices. The color set X' is $X \cup \{white\}$ where white is not a color of X. Thus, $q = 3$. White edges are precisely the edges $E(G') \setminus E(G)$. The traversal cost c' between white and every color is zero, and for any two colors of X, the traversal cost c' is the same as c. Thus, $t = 2$ and $c_{min} = 0$, implying that $k_{\min} = 0 = k$. Let $\mathcal{P}$ be a partition of the vertices $V(G)$ into pairs. G' is obtained from G by adding to G a Q_r for every pair $\{u, v\} \in \mathcal{P}$ and identifying $r-2$ of its degree one vertices and u altogether with each other as well as identifying the remaining $r-2$ vertices and v with each other. Note that the pair $\{u, v\}$ is not necessarily an edge of the graph. Every vertex of $V(G') \cap V(G)$ has $r-2$ white incident edges and at most 4 other incident edges since $\Delta(G) \leq 4$, for a total of at most $r+2$ edges. The degree of every vertex of $V(G') \setminus V(G)$ is r. Therefore, $\Delta(G') \leq r+2$. Let H' be an r-factor of G'. As the vertices of $V(G') \setminus V(G)$ are of degree exactly r in G', H' contains all the edges incident to the vertices of $V(G') \setminus V(G)$, i.e. all the white edges. By removing these edges from H' we get a 2-factor H of G. Moreover, $\mathsf{rc}(H) = \mathsf{rc}(H')$ since the white edges do not introduce a traversal cost. We conclude that (G, χ, c, k) has a 2-factor of zero cost if and only if (G', χ', c', k') has an r-factor of zero cost. □

The value of the parameter t in Theorem 1 is clearly tight. When $t = 1$, that is, when there is only one traversal cost, all the r-factors have the same reload cost. In this case the problem reduces to determining the existence of an r-factor. This problem is known to be polynomial-time solvable [23]. The following theorem states that the parameter $\Delta(G)$ of Theorem 1 is also tight.

Theorem 2. *For every $r \in \mathbb{N}_{\geq 2}$, if $\Delta(G) = r+1$, then r-MRCF can be solved in polynomial time.*

Proof. Recall that we assume $\delta(G) \geq r$. Let R be the vertices of degree r of G and $R^+ = V(G) \setminus R$ the vertices of degree $r+1$. Let also H be an r-factor of G. We observe that H is obtained by removing a perfect matching M of $G[R^+]$ from G. Every edge $e \in M$ reduces the reload cost by the sum of the traversal costs with all its incident edges. Therefore, $\mathsf{rc}(H) = \mathsf{rc}(G) - \sum_{e \in M} w(e)$ where

$$w(e) \stackrel{def}{=} \sum_{e' \in E_G(u) - e} c(e, e') + \sum_{e' \in E_G(v) - e} c(e, e')$$

for every edge $e = uv$ of G. Then, minimizing $\mathsf{rc}(H)$ boils down to the problem of finding a maximum weight perfect matching M of $G[R^+]$ with the edge weight function w, which can clearly be solved in polynomial time. □

Both the result in [11] and Theorem 1 are for general graphs and hence leave the complexity of the problem open for special graph classes. In the following (in Theorem 3), we prove hardness of r-MRCF in planar graphs even under restricted cases. We need the following Lemma.

Lemma 1. *Let G be a planar graph with an even number of vertices and $\delta(G) \geq 2$. There is a partition M of $V(G)$ into pairs such that the multigraph G' obtained by adding to G an edge between every pair of M is planar. Moreover, such a partition can be found in polynomial time.*

Proof. We construct M incrementally, starting from $M = \emptyset$. At any given point of the execution of the algorithm M is a set of vertex-disjoint pairs of vertices. We add pairs to M until $\cup M = V(G)$, implying that M is a partition of $V(G)$. A vertex v is *paired* (resp. *unpaired*) if $v \in \cup M$ (resp. $v \notin \cup M$). We use a sweep line algorithm [1] that scans the embedding from left to right. Whenever

[1] A well known technique in computational geometry [3].

the sweep line encounters a new inner face F, it pairs all the unpaired vertices of F, except possibly one vertex in case the number of unpaired vertices is odd. The pairing is done such that the pairs are non-crossing, i.e. such that one can add edges to G between the pairs while preserving planarity. Such a pairing can be easily obtained as follows. Let $v_1, v_2, \ldots, v_{2\ell}$ be the vertices to be paired, in the clockwise order they appear on the face F. The pairing, $\{\{v_1, v_2\}, \ldots, \{v_{2\ell-1}, v_{2\ell}\}\}$ is a non-crossing pairing. If the number of unpaired vertices of F is odd, the rightmost unpaired vertex of F is left unpaired.

When the sweeping phase is terminated, all the unpaired vertices are on the outer face, and their number is even. The algorithm terminates after processing the outer face exactly in the same way as an inner face was processed. □

Theorem 3. *For every $r \in [2, 5]$, r-*MRCF *is* NP*-hard even when the input graph G is planar, $\Delta(G) \leq r + 4$, $d = 2$, $k = k_{\min}$, and $q = \begin{cases} 6 & \text{if } r = 2 \\ 7 & \text{otherwise.} \end{cases}$*

Proof. We start by proving the theorem for $r = 2$ by reducing an instance (G, χ, c, k) of 2-MRCF with $\Delta(G) \leq 4$, $d = 2$, $k = k_{\min}$, $q = 2$ to an instance (G', χ', c', k') of 2-MRCF, with $\Delta(G') \leq 6$, $d = 2$, $k = k_{\min}$, $q = 6$, and G' is planar.

We rename the colors of the original graph as 1 and 2, and we add four colors $X_C = \{red, blue, green, yellow\}$. Therefore, $q = 6$. As for the cost function we use the following function c' that is an extension of c and uses only costs from $\{0, 1\}$, i.e. $d = 2$.

$$c'(\lambda, \lambda') = \begin{cases} 0 & \text{if } \lambda = \lambda' \\ 0 & \text{if } yellow \in \{\lambda, \lambda'\} \text{ and } \{1, 2\} \cap \{\lambda, \lambda'\} \neq \emptyset \\ 0 & \text{if } 1 \in \{\lambda, \lambda'\} \text{ and } 2 \notin \{\lambda, \lambda'\} \\ 1 & \text{otherwise.} \end{cases}$$

We consider a planar embedding of G where all crossings are polynomial on $|V(G)|$ and no three edges cross the same point. G' is obtained by replacing every crossing point of two edges e, e' of G by a copy of the gadget depicted in Figure 3, and 9 jokers such that one tip of each joker is identified with a distinct vertex of the gadget. We observe that $\Delta(G') \leq 6$, as required. For a gadget under consideration, the set of vertices that contain the index L (resp. R) are its *left* (resp. *right*) vertices, and the remaining vertices are its *middle* vertices. The *left part* (resp. *right part*) of the gadget is the subgraph induced by the left (resp. right) and middle vertices together. We refer to a four cycle $m_P a_{P,T} t_P a_{P,B}$ for $P \in \{L, R\}$ as a *green cycle*, to the triangle $t_T t_L t_R$ as the *middle triangle*, to a triangle that consists of an m-vertex, a t-vertex and an a-vertex as a *blue* or *red* triangle, and to the 8-cycle induced by the four c-vertices together with the four m-vertices as the *yellow rectangle*. Note that all these cycles have zero reload cost, though (despite their names) they are not monochromatic.

The two parts of the edge e (resp. e') inherit their color from e (resp. e'). Clearly, $V(G) \subseteq V(G')$. Finally, $k' = 0$.

It remains to show that G has a 2-factor with zero reload cost if and only if G' has one. Suppose that G has a 2-factor F with zero cost. We construct a 2-factor F' of G' as follows.

A part of an original edge is in F' if and only if the original edge is in F. Since F is a $2-$factor of zero cost, all the original vertices have degree 2 in F' and they incur a zero reload cost. Recall that we allow for the vertex of a gadget that is the connecting tip of a joker to be isolated in our

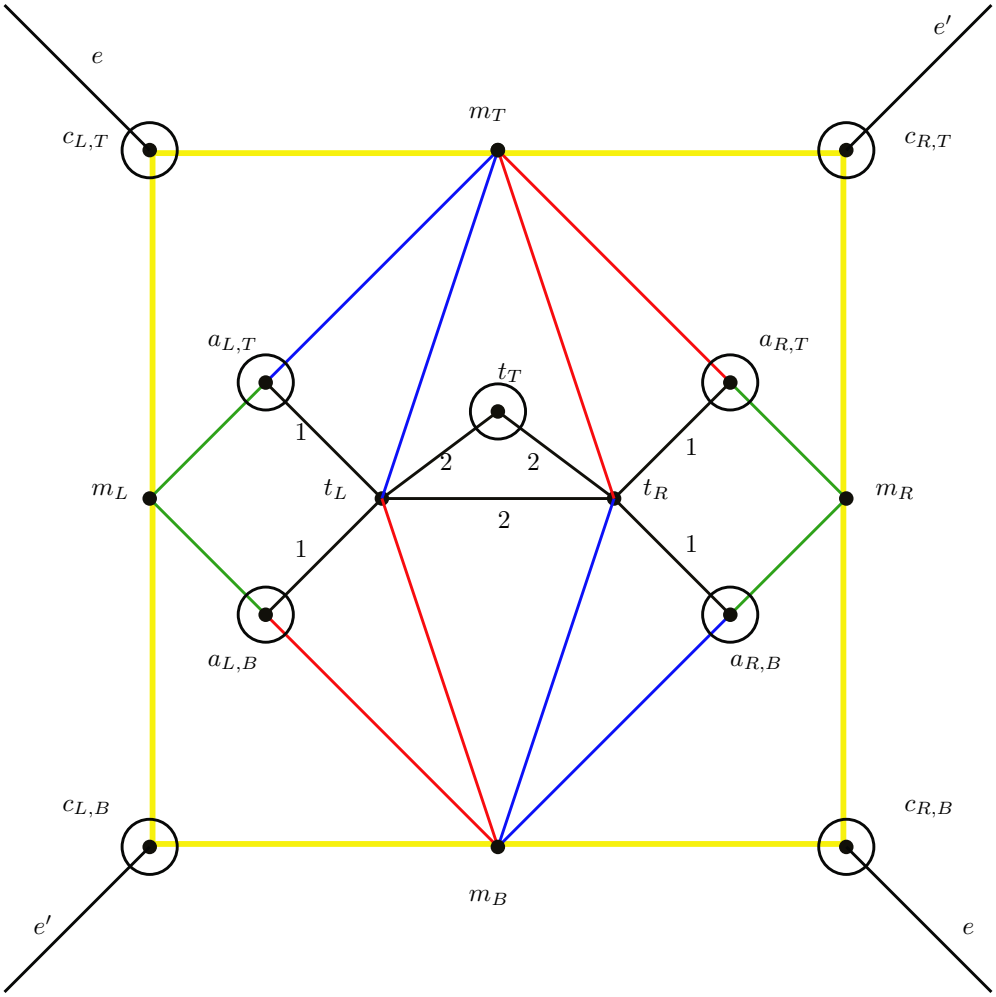


Figure 3: The planar gadget that replaces two crossing edges of G. The vertices marked with a circle are identified with a tip of a joker.

description of F' since such a vertex will be in a cycle of the joker in F'. For every gadget that corresponds to a crossing point of two edges e and e', we proceed as follows to ensure that F' incurs a zero cost in every gadget.

- $\{e, e'\} \cap F = \emptyset$: In this case F' contains the yellow rectangle and the middle triangle. Then F' incurs a zero cost in this gadget.
- $\{e, e'\} \subseteq F$: In this case F' contains the two green cycles and the two paths $c_{L,H} m_H c_{R,H}$ for $H \in \{T, B\}$. Combining with the two parts of e and e' incident to this gadget, the degree of every vertex of this gadget in F' is 2. We recall that $c'(1, yellow) = c'(2, yellow) = 0$ and conclude that F' incurs a zero cost in this gadget.
- $|\{e, e'\} \cap F| = 1$: Assume without loss of generality that $e \in F$ and $e' \notin F$. In this case F' contains the left blue triangle, the right green cycle, and the two adjacent edges of the

yellow rectangle, namely the path $c_{L,T}m_Lc_{L,B}m_Bc_{R,B}$. Combining with the two parts of e incident to this gadget, the degree of every vertex of this gadget in F' is 2. We note that $c'(\chi(e), yellow) = 0$ and conclude that F' incurs a zero cost in this gadget.

Conversely, suppose that G' has a 2-factor F' with zero cost. Recall that for any two distinct colors $\lambda, \lambda' \in \{red, blue, green, yellow\}$ we have $c'(\lambda, \lambda'') = 1$. Therefore, for every m-vertex the 2-factor F' contains two edges of the same color. If F' contains two adjacent blue (resp. red) edges, it must contain the third edge colored 1 that completes the blue (resp. red) triangle because otherwise an a-vertex incurs a positive cost, since $c'(green, red) = c'(green, blue) = 1$. If F' contains two adjacent green edges, then for the same reason, F' contains the corresponding green cycle.

Consider one of the parts, say the left part of a gadget. Since t_L is common to the blue, red and green cycles of this part, at most one of these cycles are in F'. We consider two cases:

- F' contains none of the cycles on the left part: Then the edges t_Lt_R and t_Lt_T are in F', implying that none of the cycles on the right part are in F'. Since all the m vertices have degree 2 in F' and none is covered by the inner cycles, F' must contain the yellow rectangle. Then none of the parts of e and e' are in F'.

- F' contains one cycle on the left part: In this case, since the degree of t_R must be 2 in F', F' contains one cycle of the right part. Every cycle saturates one of the m vertices, for a total of two m vertices saturated. Then, the remaining two m vertices are saturated by four yellow edges that together form two edges of the yellow rectangle. We consider two sub-cases

  - F' contains two non-adjacent edges of the yellow rectangle. In this case all the four yellow edges have degree 1. Therefore, both parts of e and both parts of e' are in F'.
  - F' contains two adjacent edges of the yellow rectangle. In this case two opposite corners of the yellow cycle, say $c_{L,T}$ and $c_{R,B}$ have one incident yellow edge, while the other two corners have either two or zero incident yellow edges. Then both parts of e (in the symmetric case, of e') are in F', and none of the parts of the other edge is in F'.

We conclude that for every edge e of G, either all its parts or none are in F'. Then F' implies a subgraph F of G. Since all the original vertices have degree 2 in F', they have degree 2 in F, i.e. F is a 2-factor of G. Furthermore, the cost at every original vertex is zero since c' extends c.

Now for any $r \in [3, 5]$, we have to reduce an instance (G, χ, c, k) of 2-MRCF where $\Delta(G) \leq 6$, $d = 2$, $k = k_{\min}$, $q = 6$, to an instance (G', χ', c', k') of r-MRCF, where $\Delta(G') \leq r + 4$, $d = 2$, $k = k_{\min}$, $q = 7$ and G' is planar. Note that, since whenever G is planar, we have $\delta(G) \leq 5$, it does not make sense to consider other values of r.

If G is odd, we subdivide an edge and add a 5-joker whose tip is identified with the newly created vertex. We color all the edges with the color of the original edge. This creates an equivalent instance having all the properties assumed for G. By Lemma 1, there is a pairing M of the vertices of G such that edges can be added between each pair of vertices while still preserving planarity. For every pair $\{u, v\} \in M$ we will add a gadget that will have the desired properties of Q_r, and in addition will preserve planarity. Namely, the gadget will have $2(r-2)$ vertices of degree one, all the remaining vertices of degree r, and will have a planar embedding in which all the degree one vertices are in the outer face. Half (i.e., $(r-2)$) of the degree one vertices of the gadget are identified with u and the remaining half are identified with v. Now every new vertex has degree r and the degree

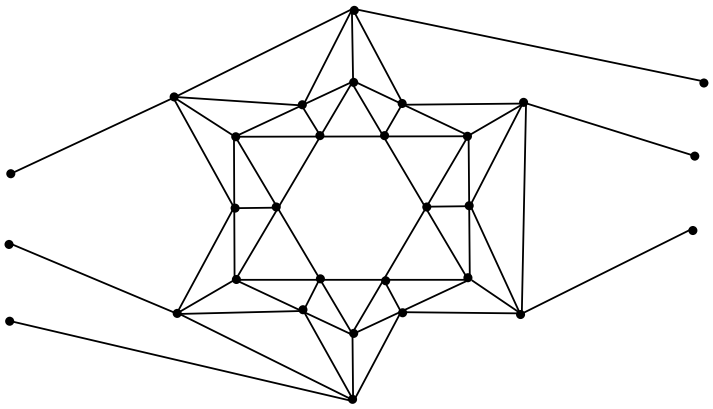


Figure 4: The planar gadget used for the case $r = 5$.

of every original vertex increased by $r - 2$. The new edges are colored with a new color λ' such that for every color λ we have $c'(\lambda', \lambda) = 0$. (G', χ', c', k') has an r-factor F' such that $\mathsf{rc}(F') = 0$ if and only if (G, χ, c, k) has a 2-factor F with $\mathsf{rc}(F) = 0$. We omit the proof of this fact since it is identical to the non-planar case.

We now describe the gadget depending on the value of r.

- $r = 3$. In this case the gadget is a Q_3 which has the desired planarity properties.
- $r = 4$. In this case the gadget is a Q_4. However, in order to get the planarity properties, the subdivided edges of the K_5 should constitute a matching.
- $r = 5$. In this case we use the gadget depicted in Figure 4.

□

4 Parameterized Complexity of r-MRCF

Lemma 2. *For every $\alpha > 1$,* 2-MRCF*, parameterized by* $\mathsf{pw}(G)$*, is* W[1]-hard *even when $d = 2$, $k = k_{\min}$, and $\overline{\mathsf{deg}}(G) < 6 \cdot \alpha$.*

Proof. The proof is by an FPT reduction from Multicolored Clique. We first note that Multicolored Clique is W[1]-hard even when the number of colors k is restricted to be odd. Indeed, the problem can be reduced to its special case as follows. Let (H, c, k) be an instance of Multicolored Clique where k is even. Let $(H', c', k + 1)$ be an instance of Multicolored Clique, where (H', c') is obtained by adding a universal vertex v colored $k + 1$ to H. It is easy to see that H contains a clique on k vertices with one vertex from each color class if and only if H' contains one. Also, by adding dummy vertices, we can also assume that each chromatic class in the input graph of Multicolored Clique has at least t vertices, where $t = \frac{\alpha}{\alpha-1}$.

Given an instance (H, c, k) of Multicolored Clique with k odd, we construct an instance (G, χ, c, k) of 2-MRCF. Let $V(H) = V_1 \uplus V_2 \uplus \cdots \uplus V_k$, where V_i is the set of vertices of $V(G)$ that are colored i, for $i \in [k]$. Notice that $|V(G) \geq t \cdot k$. Let W be an Eulerian circuit of the complete graph K_k, which exists since k is odd. We assume that $V(K_k) = [k]$. We also assume, for ease

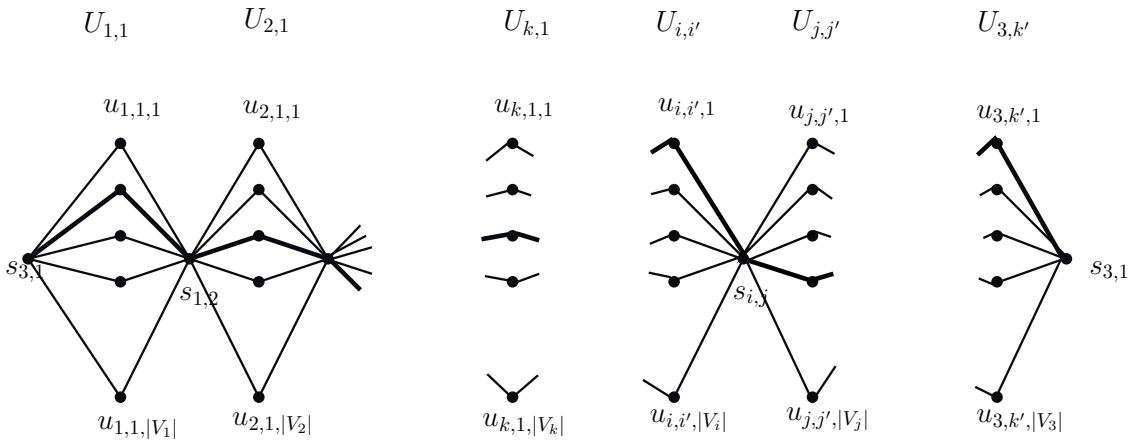


Figure 5: The $S - U$ edges of G. A sample clique cycle appears in bold.

of exposition, that W starts with the sequence of vertices $1, 2, \ldots, k$, where every vertex i of this sequence is considered as the first occurence of i in W. We also assume that the last vertex of W (i.e. the vertex before the first occurence of 1) is 3. Clearly, every $i \in [k]$ appears in W exactly $k' \stackrel{def}{=} (k-1)/2$ times. The vertex set of G is the disjoint union of three sets U, S, and T where

- U consists of k' copies $U_{i,1}, \ldots, U_{i,k'}$ of V_i for every $i \in [k]$, for a total of $\binom{k}{2}$ sets. For every $i \in [k]$, we number the vertices of V_i from 1 to $|V_i|$, and we number the vertices of every copy $U_{i,j}$ accordingly, as $u_{i,j,1}, \ldots u_{i,j,|V_i|}$.
- S consists of $\binom{k}{2}$ vertices $s_{i,j}$, one for every arc ij of W,
- $T = T_1 \cup T_2 \cup \cdots \cup T_k$, where every set T_i consists of $|V_i| - 1$ vertices $t_{i,1}, t_{i,2}, \ldots, t_{i,|V_i|-1}$.

We proceed with the description of the edge set of G, which contains three types of edges depending on their endpoints. Every edge has one endpoint in U and the other endpoint is in one of U, S or T.

- $S - U$ edges: Let $e = ij$ be an arc of W such that e is incident to the i'-th (resp. j'-th) occurence of i (resp. j) in W. Then $s_{i,j}$ is adjacent to every vertex of $U_{i,i'}$ and to every vertex of $U_{j,j'}$ (see Figure 5).
- $U - U$ edges: The $U - U$ edges form $|V(H)|$ vertex-disjoint paths, one path on k' vertices for every $v \in V(H)$. The path corresponding to the ℓ-th vertex of V_i is is $u_{i,1,\ell} u_{i,2,\ell} \ldots u_{i,k',\ell}$ (see Figure 7).
- $T - U$ edges: For every $i \in [k]$ and every $\ell \in [|V_i| - 1]$, the vertex $t_{i,\ell}$ is adjacent to $u_{i,1,\ell}$, $u_{i,1,\ell+1}$, $u_{i,k',\ell}$ and $u_{i,k',\ell+1}$ (see Figure 7).

We prove two properties of G that are necessary to conclude the lemma.

Claim 1. $\overline{\mathsf{deg}}(G) < 6\alpha$.

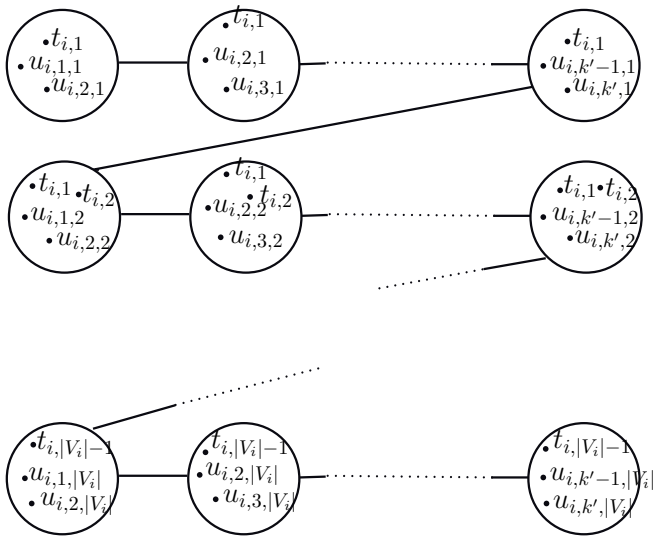


Figure 6: A path decomposition of a connected component of $G \setminus S$.

Proof of Claim 1. Clearly, $|U| = k' \cdot |V(H)|$, $|T| = |V(H)| - k$, and $S \neq \emptyset$. Therefore, $|V(G)| > (k'+1) \cdot |V(H)| - k$.

We now proceed with the calculation of $|E(G)|$. Every $U - S$ edge is incident to a vertex of U, and every vertex of U has two incident $U - S$ edges. Therefore, the number of $U - S$ edges is $2k' \cdot |V(H)|$. The total number of $U - U$ and $U - T$ edges is $(k'+1)(V(H))$, since for every vertex $v \in V(H)$ we have a cycle that contains $k'+1$ vertices, namely all the copies of v and a t vertex. Thus, $|E(G)| = (3k'+1) \cdot |V(H)|$. As $|V(G) \geq t \cdot k \Rightarrow |V(G)|/(V(G) - k) \leq t/(t-1) = \alpha$, we conclude that $\overline{\mathsf{deg}}(G)$ is

$$\frac{2|E(G)|}{|V(G)|} < \frac{2(3k'+1) \cdot |V(H)|}{(k'+1) \cdot (|V(H)| - k)} < 6 \cdot \frac{|V(H)|}{|V(H)| - k} < 6\alpha.$$

□

Claim 2. $\mathsf{pw}(G) \leq \binom{k}{2} + 3$.

Proof of Claim 2. Clearly, $\mathsf{pw}(G) \leq \mathsf{pw}(G \setminus S) + |S| = \mathsf{pw}(G \setminus S) + \binom{k}{2}$. We observe that the edge set of $G \setminus S$ consists of $U - U$ and $T - U$ edges that form connected components $T_i \cup \cup_{j'=1}^{k'} U_{i,j'}$ as depicted in Figure 7. Every such component has a path decomposition of $k' \cdot |V_i|$ bags with at most 4 vertices in a bag, as depicted in Figure 6. Therefore, $\mathsf{pw}(G \setminus S) \leq 3$ and the claim follows. □

We proceed with the description of the coloring function χ and the traversal cost function c. The color set is $V(H) \cup \{white\}$. In other words, the vertices of H corresponds to colors in the constructed instance H'; that is, there is a color in H' corresponding to each vertex in H. All the $U - U$ and $T - U$ edges are colored white, and the $S - U$ edges are colored upon their endpoint in U as follows. For every $S - U$ edge e with endpoint $u_{i,j,\ell}$, we define $\chi(e) = v_{i,\ell}$. The traversal cost

$c(v, white)$ is 1 for every $v \in V(H)$ and $c(white, white) = 0$. Finally, for every $v, v' \in V(H)$

$$c(v, v') = \begin{cases} 0 & \text{if } vv' \in E(H) \text{ or } v = v' \\ 1 & \text{otherwise.} \end{cases}$$

Finally, we set $k = 0$. This completes the construction of (G, χ, c, k), which can be clearly performed in polynomial time.

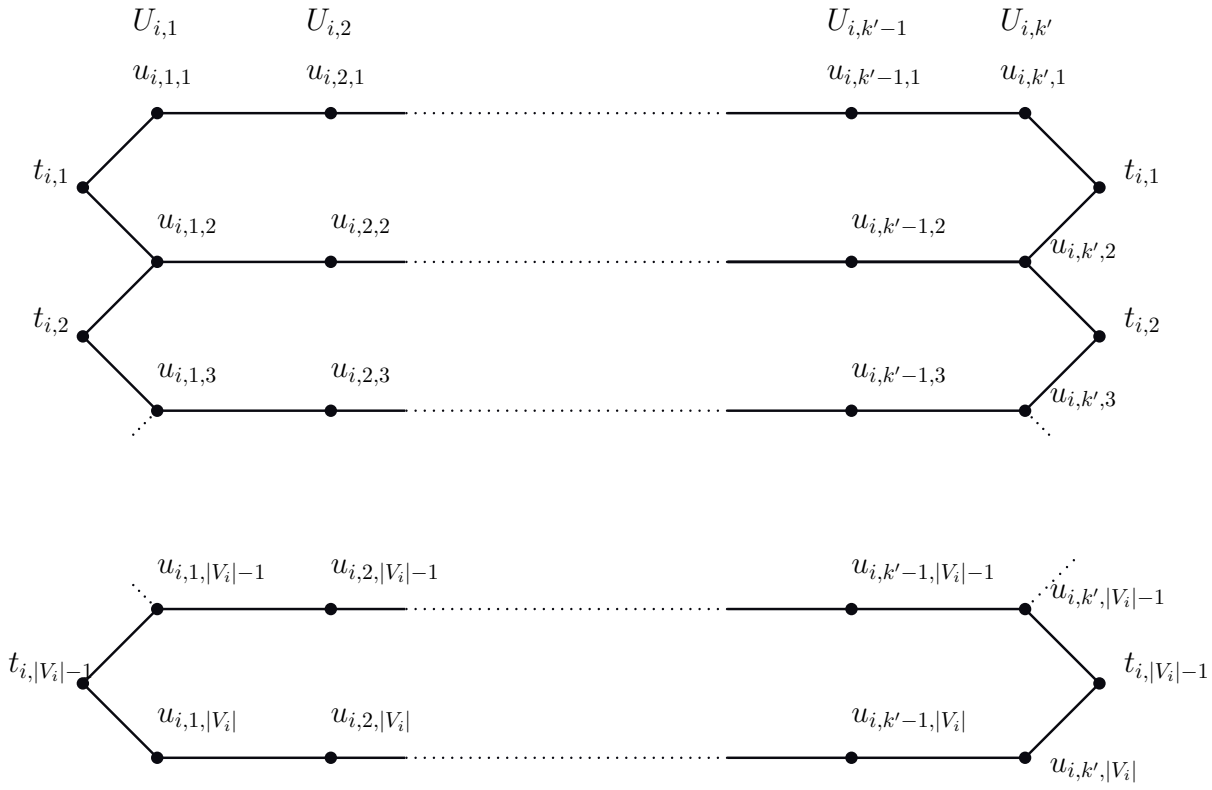


Figure 7: The $U - U$ and $T - U$ edges of G.

We now prove that H has a k-clique with one vertex from every color class if and only if G has a 2-factor F with $\mathsf{rc}(F) = 0$. Assume that H has a clique K with exactly one vertex from every color, and suppose without loss of generality that K consists of the first vertices of each color class. Let F be the 2-factor of G consisting of the following cycles.

- The clique cycle $G[C_K]$, where $C_K = S \cup \{u_{i,j,1} \mid i \in [k], j \in [k']\}$.
- The vertex cycles, one per every vertex $v \in V(H) \setminus K$. For every $i \in [k]$ and every $\ell \in [2, |V_i|]$, the vertex cycle corresponding to the vertex $v_{i,\ell} \in V(H) \setminus K$ is $G[C_{i,\ell}]$, where $C_{i,\ell} = \{t_{i,\ell-1}, u_{i,j,\ell} \mid j \in [k']\}$.

It is easy to see that every vertex is in exactly one of these cycles. Furthermore, the edges of the vertex cycles are all white, incurring a reload cost of zero. It remains to show that that the edges of the clique cycle also incur a zero cost. Indeed, the two edges incident to a vertex $u_{i,j,\ell} \in U$ incur a

cost $c(v_{i,\ell}, v_{i,\ell}) = 0$, and the two edges incident to a vertex $s_{i,j}$ incur a zero cost since the adjacent vertices are $u_{i,i',1}$ and $u_{j,j',1}$ that correspond to the vertices $v_{i,1}$ and $v_{j,1}$ of K, which are adjacent in H.

Conversely, suppose that H contains a 2-factor F with $\mathsf{rc}(F) = 0$. Every vertex $s_{i,j}$ is adjacent to two distinct vertices of U that correspond to two distinct vertices v, v' of $V(H)$. Furthermore, since the cost at vertex $s_{i,j}$ is zero, we conclude that v and v' are adjacent in H. In particular, v and v' are in different color classes. Therefore, $s_{i,j}$ is adjacent to one vertex from each of $U_{i,i'}$ and $U_{j,j'}$. The second edge incident to these vertices in F is not white, since otherwise it incurs a positive traversal cost. Thus, the other edge is also a $U - S$ edge. We conclude that the $\binom{k}{2}$ vertices S are all in one cycle C_K that also contains $\binom{k}{2}$ vertices from U, one from every set $U_{i,i'}$. Furthermore, two consecutive vertices of U in C_K correspond to two adjacent vertices of H.

All the remaining cycles must be in $G \setminus S$, which consists of k connected components $T_i \cup \cup_{i'=1}^{k'} U_{i,i'}$, for every $i \in [k]$. The result is now apparent from Figure 7 that depicts this connected component. If a vertex $u_{i,i',\ell}$ is in one of the remaining cycles, then all the vertices $u_{i,i'',\ell}$ for $i'' \in [k']$ are in the same cycle as $u_{i,i',\ell}$. Therefore, if a vertex $u_{i,i',\ell}$ is in C_K, then all the vertices $u_{i,i'',\ell}$ for $i'' \in [k']$ are in C_K. Recall that C_K contains one vertex $u_{i,i',\ell}$ from every set $U_{i,i'}$. We conclude that these vertices correspond to the same vertex $v_{i,\ell}$ of V_i. We recall that consecutive U-vertices of C_K correspond to two adjacent vertices of H. Therefore, the vertices $U \cap C_K$ correspond to the vertices of a clique of H. □

Given a graph G on n-vertices where n is even, a *pairing collection* of G is a collection M of pairs of vertices in G that forms a partition of $V(G)$. We denote $G + M = (V(G), E(G) \cup M)$. Notice that $G + M$ can be a multi-graph as the pairings in M might already be edges of G. We prove the following lemma that will prove useful in our proof.

Lemma 3. *Let G be a n-vertex graph where n is even where $\mathsf{pw}(G) \leq k$ for some $k \geq 1$. Then G has a pairing collection M such that $\mathsf{pw}(G + M) \leq k + 1$.*

Proof. We say that a vertex v of a graph is *k-simplicial* if its neighbourhood induces a complete graph on k vertices. *Proper k-paths* are recursively defined as follows: A complete graph on $k+1$ vertices is a proper k-path. A graph H on more than $k+1$ vertices is a proper k-path if it contains a k-simplicial vertex v such that the removal of v results to a proper k-path with a simplicial vertex that belongs in the neighbourhood of v in H. Notice that this recursive definition gives rice to sequence of simplicial vertices where every two successive vertices are adjacent and such that all vertices of G appear in this sequence. Therefore every proper k-tree H has a hamiltonian path, which, in case H has an even number of vertices, implies that H contains a perfect matching. It was proved in [26] (see also [19]) that every graph G of pathwidth $\leq k$ is a spanning subgraph of a proper $(k+1)$-path H. As a perfect matching in H is a pairing collection in G, the lemma follows. □

Theorem 4. *For every $r \in \mathbb{N}_{\geq 2}$, r-MRCF, parameterized by $\mathsf{pw}(G)$, is $\mathsf{W}[1]$-hard even when $d = 2$, $k = k_{\min}$ and $\overline{\mathsf{deg}}(G)$ is less than $r + \begin{cases} 4 & \text{if } r = 2 \\ 4/3 & \text{otherwise.} \end{cases}$*

Proof. For $r = 2$ the Theorem is equivalent to Lemma 2. For $r > 2$ we present an FPT reduction from the base case, i.e. $r = 2$. using the same technique as in the proof of Theorem 1. Namely, given an instance (G, χ, c, k) of 2-MRCF with $d = 2$ and $k = k_{\min} = 0$ where $\overline{\mathsf{deg}}(G) < 6$, we

construct an instance (G', χ', c', k') of r-MRCF where $d = 2$ and the average degree of G' is less than $r + 4/3$ by adding to G a pair collection, and replacing every new edge by the gadget Q_r. By the last part of the proof of Theorem 1, G' has an r-factor of zero cost if and only if G has a 2-factor of zero cost.

Claim 3. *The average degree of G' is at most $r + 4/3$.*

Proof of Claim 3. Let $n = |V(G)|$. Then $|E(G)| < 3n$. On the other hand, the number of edges added by every Q_r is $\frac{r(r+1)}{2} + r - 2$. Summarizing, we have

$$\begin{aligned} |E(G')| &< 3n + \frac{n}{2}\left(\frac{r(r+1)}{2} + r - 2\right) \\ |V(G')| &= n + \frac{n}{2}(r+1). \end{aligned}$$

We conclude that the average degree of G' is less than

$$\frac{6 + \left(\frac{r(r+1)}{2} + r - 2\right)}{1 + \frac{r+1}{2}} = \frac{r^2 + 3r + 8}{r+3} = r + \frac{8}{r+3} \leq r + \frac{4}{3}.$$

□

Claim 4. $\mathsf{pw}(G') \leq (\mathsf{pw}(G) + 1) \cdot \frac{r+1}{r-1} - 1$.

Proof of Claim 4. According to Lemma 3, there is a pairing collection M of G such that $\mathsf{pw}(G + M) \leq \mathsf{pw}(G) + 1$.

The operation of replacing each pair of M by a Q_r can be reflected in the path decomposition as follows. Each bag has $k+1$ vertices and among the edges with endpoints in this bag, at most $(k+1)/2$ of them can be pairs that belong in M. This means that after the insertion of at most $(k+1)/2$ copies of Q_r in each bag the new path decomposition will have width at most $k + 1 + (k+1)/2 \cdot (r+1) = (\mathsf{pw}(G) + 1) \cdot \frac{r+1}{r-1}$. □

Finally, we note that r is a constant of the problem. Therefore, the function proven in Claim 4, is a function of $\mathsf{pw}(G)$ as required. □

We conclude with the following algorithmic result.

Theorem 5. *For every $r \in \mathbb{N}_{\geq 2}$, r-MRCF, parameterized by $\min\{q, \Delta(G)\}$ and $\mathsf{tw}(G)$ is in FPT. Specifically, it can be solved by an algorithm that runs in time $\mathcal{O}^*\left(\binom{\min\{q,\Delta(G)\}+r}{r}^{2(\mathsf{tw}(G)+1)}\right)$.*

Proof. We assume that the graph G is given together with a nice pair $(\mathcal{D}, \mathcal{G})$ where $\mathcal{D} = (T, \mathcal{X}, r)$. Indeed, by using for instance the algorithm of Bodlaender *et al.* [4], we can compute in time $2^{\mathcal{O}(\mathsf{tw}(G))} \cdot n$ a tree decomposition of G of width at most $5\mathsf{tw}(G)$. Note that this running time is clearly dominated by the running time stated in Theorem 5. Recall also that, by [7, 20], given a tree decomposition, it can be transformed in polynomial time into a nice pair.

We first introduce some notation. Let $t \in T$, $v \in X_t$, and F an r-factor of G. The graph $F_t = F[V(G_t)]$ is a partial r-factor of G. We encode the edges $E_{G_t}(v)$ incident to v in G_t as a vector $\mathbf{E}_{t,v}$ of non-negative integers indexed by the colors of the instance. The value of the entry

λ of $\mathbf{E}_{t,v}$ is the number of edges of $E_{G_t}(v)$ colored λ. Clearly, the sum of the entries of this vector, denoted by $\|\mathbf{E}_{t,v}\|$ is $|E_{G_t}(v)|$. The encoding of a single edge e colored $\chi(e) = \lambda$, incident to v is the vector $\mathbf{u}_\lambda$ that has a 1 in entry λ and zero elsewhere. We now define the set of all vectors encoding a subset of $|E_{G_t}(v)$ consisting of at most r edges. In other words, these are the sets of edges incident to v in G_t that can be part of an r-factor of G:

$$\mathcal{E}_{t,v} \stackrel{def}{=} \{\mathbf{E} \mid \mathbf{E} \leq \mathbf{E}_{t,v}, \|\mathbf{E}\| \leq r\}.$$

Note that $|\mathcal{E}_{t,v}| \leq \binom{q+r}{r}$. Let $\mathbf{E}, \mathbf{E}' \in \mathcal{E}_{t,v}$ encode two disjoint sets of edges incident to v. Each of $\mathbf{E}, \mathbf{E}'$ has an associated reload cost on v. The reload cost of their union $\mathbf{E} + \mathbf{E}'$ is the sum of their individual reload costs, plus the reload costs incurred by the traversals between the two sets, which is $c_v(\mathbf{E}, \mathbf{E}') \stackrel{def}{=} \sum_\lambda \sum_{\lambda'} c(\lambda, \lambda') \cdot \mathbf{E}_\lambda \cdot \mathbf{E}_{\lambda'}$.

Algorithm 1 DynProg

Require: An instance (G, χ, c, k) of r-MRCF
Require: A nice pair $\mathcal{D}, \mathcal{G}$ where $\mathcal{D} = (T, r, \mathcal{X})$ is a tree decomposition of G.

for $t \in T$ (using a bottom-up traversal) **do**
  $\mathcal{T}_k(\mathbf{E_t}) \leftarrow \infty$ for every $\mathbf{E_t} \in \mathcal{E}_t$.
  if $t = r$ **then**
    return $\mathcal{T}_r(()) \leq k$. ▷ $T_r = \emptyset, \mathcal{E}_r = \{()\}$
  else if t is a leaf node **then**
    $T_t(()) = 0$.
  else if t is a join node with children t', t'' **then**
    for $\mathbf{E_{t'}} \in \mathcal{E}_{t'}$ **do**
      for $\mathbf{E_{t''}} \in \mathcal{E}_{t''}$ **do**
        if $\mathbf{E_{t'}} + \mathbf{E_{t''}} \in \mathcal{E}_t$ **then**
          $cost \leftarrow \mathcal{T}_{t'}(\mathbf{E_{t'}}) + \mathcal{T}_{t''}(\mathbf{E_{t''}}) + c_{X_t}(\mathbf{E_{t'}}, \mathbf{E_{t''}})$.
          if $cost < \mathcal{T}_t(\mathbf{E_{t'}} + \mathbf{E_{t''}})$ **then** $\mathcal{T}_t(\mathbf{E_{t'}} + \mathbf{E_{t''}}) \leftarrow cost$
  else if t introduces the vertex v and has child t' **then**
    for $\mathbf{E_{t'}} \in \mathcal{E}_{t'}$ **do**
      $\mathcal{T}_t((\mathbf{0}, \mathbf{E_{t'}})) = \mathcal{T}_{t'}(\mathbf{E_{t'}})$. ▷ w.l.o.g. v is the first vertex of X_t.
  else if t introduces the edge $e = uv$ and has child t' **then** ▷ $\mathcal{E}_{t'} \subseteq \mathcal{E}_t$.
    for $\mathbf{E_t} \in \mathcal{E}_{t'}$ **do** ▷ Partial r-factors that do not contain e
      $\mathcal{T}_t(\mathbf{E_t}) \leftarrow \mathcal{T}_{t'}(\mathbf{E_t})$
    for $\mathbf{E_t} \in \mathcal{E}_t \setminus \mathcal{E}_{t'}$ **do** ▷ Partial r-factors that contain e
      $\mathbf{E_{t'}} \leftarrow \mathbf{E_t} - (\mathbf{u}_{\chi(\mathbf{e})}, \mathbf{u}_{\chi(\mathbf{e})}, \mathbf{0}, \ldots, \mathbf{0})$
      $\mathcal{T}_t(\mathbf{E_t}) \leftarrow \mathcal{T}_{t'}(\mathbf{E_{t'}}) + c_u(\mathbf{E_{t'}}, \mathbf{u}_{\chi(\mathbf{e})}) + c_v(\mathbf{E_{t'}}, \mathbf{u}_{\chi(\mathbf{e})})$
  else ▷ t is a forget node that forgets v and has child t'
    for $\mathbf{E_{t'}} = (\mathbf{e_v}, \mathbf{E_t}) \in \mathcal{E}_{t'}$ **do**
      if $\|\mathbf{e_v}\| = r$ and $\mathcal{T}_{t'}(\mathbf{E_{t'}}) < \mathcal{T}_t(\mathbf{E_t})$ **then**
        $\mathcal{T}_t(\mathbf{E_t}) \leftarrow \mathcal{T}_{t'}(\mathbf{E_{t'}})$

We extend this definition to any subset X of vertices as $c_X(\mathbf{E}, \mathbf{E}') \stackrel{def}{=} \sum_{x \in X} c_x(\mathbf{E}, \mathbf{E}')$.

$\mathcal{E}_t$ is the cartesian product $\times_{v \in X_t} \mathcal{E}_{t,v}$, i.e. every element of $\mathcal{E}_t$ is a vector of $|X_t|$ vectors that encodes a set of edges incident to X_t that is possibly part of an r-factor of G. Clearly, $|\mathcal{E}_t| \leq \binom{q+r}{r}^{|X_t|} \leq \binom{q+r}{r}^{w+1}$. For every $t \in T$ we maintain a function $\mathcal{T}_t : \mathcal{E}_t \to \mathbb{N}_{\geq 0}$ that returns for any given $\mathbf{E_t} \in \mathcal{E}_t$, the minimum cost of a partial r-factor whose edges incident to X_t are encoded by $\mathbf{E_t}$.

The dynamic programming algorithm whose pseudo code is given in Algorithm 1 performs a bottom-up traversal of T and updates the tables $\mathcal{T}_t$ at every node, and finally returns the result by inspecting the value $\mathcal{T}_r(())$ (recall that $\mathcal{E}_r = \{()\}$.

As for the running time of the algorithm, we observe that the dominant running time for a node is obtained in the case of a join node, in which case the running time is $\mathcal{O}(|\mathcal{E}_{t'}| \cdot |\mathcal{E}_{t''}|) = \mathcal{O}^*\left(\binom{q+r}{r}^{2(\mathsf{tw}(G)+1)}\right)$, ignoring polynomial factors. Alternatively, we can encode the vectors of $\mathcal{E}_{t,v}$ using a binary vector of length $\Delta(G)$. In this case $|\mathcal{E}_t| = \binom{\Delta(G)+r}{r}$, implying the running time in the statement of the Theorem. □

References

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

[2] Stamatios Arkoulis, Evangelos Anifantis, Vasileios Karyotis, Symeon Papavassiliou, and Nikolaos Mitrou. On the optimal, fair and channel-aware cognitive radio network reconfiguration. *Computer Networks*, 57(8):1739–1757, 2013.

[3] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

[4] Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.

[5] Abdulkadir Celik and Ahmed E Kamal. Green cooperative spectrum sensing and scheduling in heterogeneous cognitive radio networks. *IEEE Transactions on Cognitive Communications and Networking*, 2(3):238–248, 2016.

[6] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. URL: http://dx.doi.org/10.1007/978-3-319-21275-3, doi:10.1007/978-3-319-21275-3.

[7] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proc. of the 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 150–159. IEEE Computer Society, 2011.

[8] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[9] Giulia Galbiati. The complexity of a minimum reload cost diameter problem. *Discrete Applied Mathematics*, 156(18):3494–3497, 2008.

[10] Giulia Galbiati, Stefano Gualandi, and Francesco Maffioli. On minimum changeover cost arborescences. In *Proc. of the 10th International Symposium on Experimental Algorithms (SEA)*, volume 6630 of *LNCS*, pages 112–123, 2011.

[11] Giulia Galbiati, Stefano Gualandi, and Francesco Maffioli. On minimum reload cost cycle cover. *Discrete Applied Mathematics*, 164:112–120, 2014.

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

[13] Laurent Gourvès, Adria Lyra, Carlos Martinhon, and Jérôme Monnot. The minimum reload s-t path, trail and walk problems. *Discrete Applied Mathematics*, 158(13):1404–1417, 2010.

[14] D. Gözüpek, S. Özkan, C. Paul, I. Sau, and M. Shalom. Parameterized complexity of the MINCCA problem on graphs of bounded decomposability. In *Proc. of the 42nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 9941 of *LNCS*, pages 195–206, 2016. Full version available at `arXiv:1509.04880`.

[15] Didem Gözüpek, Seyed Buhari, and Fatih Alagöz. A spectrum switching delay-aware scheduling algorithm for centralized cognitive radio networks. *IEEE Transactions on Mobile Computing*, 12(7):1270–1280, 2013.

[16] Didem Gözüpek, Hadas Shachnai, Mordechai Shalom, and Shmuel Zaks. Constructing minimum changeover cost arborescenses in bounded treewidth graphs. *Theoretical Computer Science*, 621:22–36, 2016.

[17] Didem Gözüpek and Mordechai Shalom. Edge coloring with minimum reload/changeover costs. *arXiv preprint arXiv:1607.06751*, 2016.

[18] Didem Gözüpek, Mordechai Shalom, Ariella Voloshin, and Shmuel Zaks. On the complexity of constructing minimum changeover cost arborescences. *Theoretical Computer Science*, 540:40–52, 2014. URL: `http://dx.doi.org/10.1016/j.tcs.2014.03.023`, `doi:10.1016/j.tcs.2014.03.023`.

[19] Haim Kaplan and Ron Shamir. Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques. 25(3):540–561, 1996.

[20] T. Kloks. *Treewidth. Computations and Approximations*. Springer-Verlag LNCS, 1994.

[21] Henk Meijer, Yurai Núñez-Rodríguez, and David Rappaport. An algorithm for computing simple k-factors. *Information Processing Letters*, 109(12):620–625, 2009.

[22] Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences*, 67(4):757–771, 2003. URL: `http://dx.doi.org/10.1016/S0022-0000(03)00078-3`, `doi:10.1016/S0022-0000(03)00078-3`.

[23] R. Pulleyblank. *Faces of Matching Polyhedra*. PhD thesis, University of Waterloo, 1973.

[24] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. ACM. URL: http://doi.acm.org/10.1145/800133.804350, doi:10.1145/800133.804350.

[25] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

[26] Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. Mixed searching and proper-path-width. *Theoretical Computer Science*, 137(2):253 – 268, 1995. URL: http://www.sciencedirect.com/science/article/pii/030439759400160K, doi:10.1016/0304-3975(94)00160-K.

[27] Hans-Christoph Wirth and Jan Steffan. Reload cost problems: minimum diameter spanning tree. *Discrete Applied Mathematics*, 113(1):73–85, 2001.