On the Parameterized Complexity of Simultaneous Deletion Problems

Akanksha Agrawal[1], R. Krithika[2], Daniel Lokshtanov[1], Amer E. Mouawad[1], and M. S. Ramanujan[3]

1 **Department of Informatics, University of Bergen, Norway**
`{akanksha.agrawal|a.mouawad}@uib.no`, `daniello@ii.uib.no`
2 **Institute of Mathematical Sciences, HBNI, Chennai, India**
`rkrithika@imsc.res.in`
3 **Algorithms and Complexity Group, Vienna University of Technology, Austria**
`ramanujan@ac.tuwien.ac.at`

Abstract

For a family of graphs $\mathcal{F}$, an n-vertex graph G, and a positive integer k, the $\mathcal{F}$-Deletion problem asks whether we can delete at most k vertices from G to obtain a graph in $\mathcal{F}$. $\mathcal{F}$-Deletion generalizes many classical graph problems such as Vertex Cover, Feedback Vertex Set, and Odd Cycle Transversal. A (multi) graph $G = (V, \cup_{i=1}^{\alpha} E_i)$, where the edge set of G is partitioned into α color classes, is called an α-edge-colored graph. A natural extension of the $\mathcal{F}$-Deletion problem to edge-colored graphs is the Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion problem. In the latter problem, we are given an α-edge-colored graph G and the goal is to find a set S of at most k vertices such that each graph $G_i - S$, where $G_i = (V, E_i)$ and $1 \le i \le \alpha$, is in $\mathcal{F}_i$. Recently, a subset of the authors considered the aforementioned problem with $\mathcal{F}_1 = \ldots = \mathcal{F}_\alpha$ being the family of all forests. They showed that the problem is fixed-parameter tractable when parameterized by k and α and can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(\alpha k)})$ time[1]. In this work, we initiate the investigation of the complexity of Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion with different families of graphs. In the process, we obtain a complete characterization of the parameterized complexity of this problem when one or more of the $\mathcal{F}_i's$ is the class of bipartite graphs and the rest (if any) are forests. We show that if $\mathcal{F}_1$ is the family of all bipartite graphs and each of $\mathcal{F}_2 = \mathcal{F}_3 = \ldots = \mathcal{F}_\alpha$ is the family of all forests then the problem is fixed-parameter tractable parameterized by k and α. However, even when $\mathcal{F}_1$ and $\mathcal{F}_2$ are both the family of all bipartite graphs, then the Simultaneous $(\mathcal{F}_1, \mathcal{F}_2)$-Deletion problem itself is already W[1]-hard.

1998 ACM Subject Classification G.2.2 Graph Algorithms, I.1.2 Analysis of Algorithms

Keywords and phrases parameterized complexity, feedback vertex set, odd cycle transversal, edge-colored graphs, simultaneous deletion

Digital Object Identifier 10.4230/LIPIcs..2016.

1 Introduction

Given their tremendous modelling power, graphs have become an integral part of theoretical computer science in general, and of algorithm design in particular. One graph problem which encapsulates many problems of both practical and theoretical interest is $\mathcal{F}$-Deletion. For a family of graphs $\mathcal{F}$, an n-vertex graph G, and a positive integer k, the $\mathcal{F}$-Deletion problem asks whether we can delete at most k vertices from G to obtain a graph in $\mathcal{F}$.

[1] We use the $\mathcal{O}^\star$ notation which hides factors that are polynomial in the input size.

Editors: John Q. Open and Joan R. Acces; Article No. ; pp. :1–:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

To state a few, $\mathcal{F}$-Deletion generalizes problems such as Vertex Cover [6], Feedback Vertex Set (FVS) [5, 8, 17], Vertex Planarization [15], Odd Cycle Transversal (OCT) [22, 14, 18], Interval Vertex Deletion [3], Chordal Vertex Deletion [4], and Planar $\mathcal{F}$-Deletion [11, 16].

A graph $G = (V, \cup_{i=1}^{\alpha} E_i)$, where the edge set of G is partitioned into α color classes, is called an α-edge-colored graph. Edge-colored graphs are fundamental in graph theory and have been extensively studied in the literature, especially for alternating cycles and monochromatic subgraphs [2]. A natural extension of the $\mathcal{F}$-Deletion problem to edge-colored graphs is the Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion problem. In the latter problem, we are given an α-edge-colored graph G and the goal is to find a set S of at most k vertices such that each graph $G_i - S$ is in $\mathcal{F}_i$, where $G_i = (V, E_i)$ and $1 \leq i \leq \alpha$. Recently, Cai and Ye [2] studied several problems restricted to 2-edge-colored graphs, where edges are colored either red or blue. They asked, as an open question, whether the Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion problem parameterized by k, with $\alpha = 2$ and $\mathcal{F}_1 = \mathcal{F}_2$ being the family of all forests, is fixed-parameter tractable (FPT), i.e. whether the problem can be solved in $\mathcal{O}^\star(f(k))$ time [10] (for some computable function f). Agrawal et al. [1] and Ye [24] answered this question in the affirmative. In particular, it was shown in [1] that the problem can be solved by an algorithm running in $\mathcal{O}^\star(2^{\mathcal{O}(\alpha k)})$ time. This work pointed to a few natural further directions for research. For instance, does Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion remain fixed-parameter tractable when the family of all forests is replaced by the family of all bipartite graphs? What is the complexity of the problem when not all families are equal?

The results in this work allow us to take a significant step towards a better understanding of simultaneous deletion problems in general. To that end, we investigate the complexity of Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion in two settings. First, we consider the problem with $\mathcal{F}_1$ being the family of all bipartite graphs and $\mathcal{F}_2 = \mathcal{F}_3 = \ldots = \mathcal{F}_\alpha$ being the family of all forests. We call this problem Simultaneous FVS/OCT and define it as follows.

> Simultaneous FVS/OCT **Parameter(s):** k and α
> **Input:** An α-edge-colored graph $G = (V, \cup_{i=1}^{\alpha} E_i)$ and an integer k.
> **Question:** Is there a set $S \subseteq V$ of size at most k such that $G_1 - S$ is a bipartite graph and $G_2 - S$, ..., $G_\alpha - S$ are acyclic, where $G_i = (V, E_i)$ and $1 \leq i \leq \alpha$?

We call a solution S to the Simultaneous FVS/OCT problem a *sim-fvs-oct*. Our first contribution is an algorithm that, given an instance $(G = (V, \cup_{i=1}^{\alpha} E_i), k)$ of Simultaneous FVS/OCT, runs in time $\mathcal{O}^\star(k^{\mathsf{poly}(\alpha,k)})$ and either computes a sim-fvs-oct in G of size at most k or correctly concludes that such a set does not exist.

In the second setting, we consider the Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion problem where $\mathcal{F}_1 = \ldots = \mathcal{F}_\alpha$ is the family of all bipartite graphs. We call this problem Simultaneous OCT and define it as follows.

> Simultaneous OCT **Parameter(s):** k and α
> **Input:** An α-edge-colored graph $G = (V, \cup_{i=1}^{\alpha} E_i)$ and an integer k.
> **Question:** Is there a set $S \subseteq V$ of size at most k such that $G_i - S$ is bipartite, where $G_i = (V, E_i)$ and $1 \leq i \leq \alpha$?

We refer to a solution S to the Simultaneous OCT problem as a *sim-oct*. Our second (and rather surprising) contribution is a negative answer to the first open question of Agrawal et al. [1]. We show that, even for $\alpha = 2$, the Simultaneous OCT problem is W[1]-hard. To prove this result, we first reduce the well-known Multicolored Clique problem [7] to an auxiliary problem we call Simultaneous Cut. Simultaneous Cut is a natural generalization of the classical (s,t)-CUT problem to edge-colored graphs. Finally, we show that Simultaneous Cut can be reduced to Simultaneous OCT. Notice that W[1]-hardness of Simultaneous OCT implies that Simultaneous $(\mathcal{F}_1, \ldots, \mathcal{F}_\alpha)$-Deletion problem with at least two of the families being the family of all bipartite graphs is W[1]-hard.

Overview of the algorithm. Note that for any fixed k and α, our algorithm for solving the Simultaneous FVS/OCT problem runs in polynomial time. The said algorithm can be broken down into four stages, three of which are reductions to auxiliary problems. Initially, as was first proposed by Ye [24], we use the notion of compact representations of feedback vertex sets (see Section 2 for formal definitions) to reduce Simultaneous FVS/OCT into $2^{\mathcal{O}(\alpha k)}$ instances of the Colorful OCT problem, which is formally defined as follows. We note that, in any reduced instance, ℓ will be bounded above by αk.

> Colorful OCT **Parameter(s):** k and ℓ
> **Input:** A graph $G = (V, E)$, integers k and ℓ, and a grouping $\mathcal{P}$ of the vertices of G into (not necessarily distinct) sets $\{P_1, \ldots, P_\ell\}$.
> **Question:** Is there a set $S \subseteq V$ of size at most k such that $G - S$ is a bipartite graph and $S \cap P_i \neq \emptyset$, for $i \in \{1, \ldots, \ell\}$?

Intuitively, compact representations give us a partition of a vertex subset of the graph into sets such that picking one vertex from each part is "guaranteed" to constitute a feedback vertex set of each graph G_i, $2 \leq i \leq \alpha$. As such, we are able to encode the feedback vertex set "side" of the Simultaneous FVS/OCT problem (via the reduction) as colors on the vertices (i.e. different sets in $\mathcal{P}$ represent different colors for each vertex) and focus on a "colored" variant of Odd Cycle Transversal. Naturally, the second stage is to solve the Colorful OCT problem within the claimed running time. To do so, we reduce an instance of Colorful OCT to an instance of the compression variant of the problem, i.e. Colorful OCT Compression. This problem assumes an odd cycle transversal of size at most k as part of the input. Note that finding an odd cycle transversal of a graph $G = (V, E)$ of size at most k can be accomplished using the fixed-parameter tractable algorithms for OCT parameterized by solution size [14, 22], both of which run in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time.

> Colorful OCT Compression **Parameter(s):** k and ℓ
> **Input:** A graph $G = (V, E)$, integers k and ℓ, a grouping $\mathcal{P}$ of the vertices of G into (not necessarily distinct) sets $\{P_1, \ldots, P_\ell\}$, and a set $O \subseteq V(G)$ of size at most k such that $G - O$ is bipartite.
> **Question:** Is there a set $S \subseteq V$ of size at most k such that $G - S$ is a bipartite graph and $S \cap P_i \neq \emptyset$, for $i \in \{1, \ldots, \ell\}$?

Now, to solve an instance of Colorful OCT Compression, we reduce it into $2^{\mathcal{O}(k)}$ instances of yet another problem, namely Colorful Separator. This reduction is in many ways similar to the iterative compression algorithm for solving the Odd Cycle Transversal problem [7, 13, 23].

> Colorful Separator **Parameter(s):** k and ℓ
> **Input:** A graph $G = (V, E)$, integers k and ℓ, a grouping $\mathcal{P}$ of the vertices of G into (not necessarily distinct) sets $\{P_1, \ldots, P_\ell\}$, and vertices s and t in $V(G)$.
> **Question:** Is there an (s,t)-separator $S \subseteq V \setminus \{s, t\}$ such that $|S| \leq k$ and $S \cap P_i \neq \emptyset$, for each $i \in \{1, \ldots, \ell\}$?

Finally, and arguably the most technical part of our algorithm, is to show how to solve an instance of Colorful Separator. We will in fact solve a much more general problem, which we define in Section 4 (to keep the presentation clear). Our two main ingredients are a dynamic programming routine and a generalization of the concept of important separators, which has been recently defined to design parameterized algorithms for several "cut" problems [12, 19, 20]. We note that an alternative algorithm for solving Colorful Separator can be obtained by applying the treewidth reduction result of Marx et al. [21]. However, a "simple" application of this result would give an algorithm with a worse running time (double exponential).

2 Preliminaries

We denote the set of natural numbers by $\mathbb{N}$. For $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, 2, \ldots, n\}$. Given a universe $\mathcal{U}$, a set $S \subseteq \mathcal{U}$, and a family of sets $\mathcal{F} = \{F_1, \ldots, F_\ell\}$ over $\mathcal{U}$, we let $\mathcal{F}\big|_S$ denote the *restriction* of F to S, i.e. $\mathcal{F}\big|_S = \{F_1 \cap S, \ldots, F_\ell \cap S\}$. We use standard terminology from the book of Diestel [9] for the graph-related terms which are not explicitly defined here. For a graph G, we use $V(G)$ and $E(G)$ to denote the vertex and edge sets of G, respectively. For $S \subseteq V(G)$, by $N_G(S)$ we denote the set $\{u \in V(G) \setminus S \mid (u,v) \in E(G) \wedge v \in S\}$. We drop the subscript G from $N_G(S)$ when the context is clear. For a vertex subset $S \subseteq V(G)$, by $G[S]$ we denote the graph with vertex set S and edge set $\{(u,v) \in E(G) \mid u, v \in S\}$. By $G - S$ we denote the graph $G[V(G) \setminus S]$. A *path* from v_1 to v_ℓ in a graph G is a sequence of vertices $v_1, v_2, \ldots, v_\ell$ such that for all $i \in [\ell - 1]$, $(v_i, v_{i+1}) \in E(G)$. We call such a path a (v_1, v_ℓ)-path. For $X, Y \subseteq V(G)$, an (X,Y)-path in G is a path $v_1, v_2, \ldots, v_\ell$ such that $v_1 \in X$ and $v_\ell \in Y$. We say that X and Y are *linked* in G if there exists an (X,Y)-path in G. We say that vertices in Y are *reachable* from X if, for all $y \in Y$, there exists $x \in X$ such that there is a path from x to y.

A vertex subset $S \subseteq V(G)$ is a *feedback vertex set (fvs)* in G if $G - S$ is a forest. If there is no $S' \subset S$ such that $G - S'$ is a forest then S is a *minimal feedback vertex set (minimal fvs)* in G. A vertex subset $S \subseteq V(G)$ is an *odd cycle transversal (oct)* in G if $G - S$ is bipartite. If there is no $S' \subset S$ such that $G - S'$ is a bipartite graph then S is a *minimal odd cycle transversal (minimal oct)* in G. For a graph G and set $X \subseteq V(G)$, we refer to a partition (A, B) of X as a *valid bipartition* of $G[X]$ if $G[A]$ and $G[B]$ are both edgeless graphs. We refer to a valid bipartition of $V(G)$ as a valid bipartition of the graph G.

▶ **Definition 2.1.** Let G be a graph and X and Y be disjoint subsets of $V(G)$. A vertex set S disjoint from $X \cup Y$ is called an *(X,Y)-separator* if there is no (X,Y)-path in $G - S$. We denote by $R_G(X,S)$ the set of vertices of $G - S$ reachable from vertices of X via paths and by $NR_G(X,S)$ the set of vertices of $G - S$ *not* reachable from vertices of X.

We remark that it is not necessary that Y and $N(X)$ be disjoint in the above definition. If these sets do intersect, then there is no (X,Y)-separator in the graph.

▶ **Definition 2.2.** [13] A *compact representation* of a set $\mathcal{S}$ of minimal feedback vertex sets of a graph G is a collection $\mathcal{C}$ of pairwise disjoint subsets of $V(G)$ such that choosing exactly one vertex from every set in $\mathcal{C}$ results in a minimal feedback vertex set for G that is in $\mathcal{S}$.

▶ **Lemma 2.3.** [13] *The set of all minimal feedback vertex sets of size at most k can be represented by a collection of compact representations of size $2^{\mathcal{O}(k)}$. Furthermore, given a graph $G = (V, E)$ and a feedback vertex set F for G of size $k + 1$, we can enumerate the compact representations of all minimal feedback vertex sets for G having size at most k in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time.*

3 From Simultaneous FVS/OCT to Colorful OCT

We first describe how to reduce an instance of Simultaneous FVS/OCT to $2^{\mathcal{O}(\alpha k)}$ instances of Colorful OCT. Note that since both Feedback Vertex Set [20] and Odd Cycle Transversal [22, 14] can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time, we assume that, along with an instance $(G = (V, \cup_{i=1}^{\alpha} E_i), k)$, we are given sets $O, F_2, \ldots, F_\alpha \subseteq V(G)$ of size at most k such that $G_1 - O$ is a bipartite graph and $G_i - F_i$, $2 \leq i \leq \alpha$, is acyclic (as otherwise we can safely conclude that the given instance is a no-instance).

▶ **Lemma 3.1.** *There is an algorithm that, given an instance $(G = (V, \cup_{i=1}^{\alpha} E_i), k)$ of* Simultaneous FVS/OCT*, runs in time $\mathcal{O}^\star(2^{\mathcal{O}(\alpha k)})$ and returns a set of $2^{\mathcal{O}(\alpha k)}$ instances of* Colorful OCT *such that the original instance is a yes-instance if and only if at least one of the returned instances is a yes-instance.*

Proof. Armed with the sets F_i which are of size at most k, we apply the algorithm of Lemma 2.3 to each graph G_i, $2 \leq i \leq \alpha$, to obtain a set of compact representations $\mathbf{C}_i = \{\mathcal{C}_i^1, \mathcal{C}_i^2, \ldots\}$, $2 \leq i \leq \alpha$. Note that each $\mathbf{C}_i$ is of size $2^{\mathcal{O}(k)}$ and each $\mathcal{C}_i^j$ is of size at most k. The said algorithm runs in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time for each graph G_i. For each tuple $\{\mathcal{C}_2^{j_2}, \ldots, \mathcal{C}_\alpha^{j_\alpha}\} \in \mathbf{C}_2 \times \ldots \times \mathbf{C}_\alpha$, we construct an instance $(G', \mathcal{P}, k', \ell)$ of Colorful OCT as follows. We let $G' = (V, E_1)$, $k' = k$, and $\ell = \sum_{i=2}^{\alpha} |\mathcal{C}_i^j| \leq \alpha k$.

For each $\mathcal{C} \in \{\mathcal{C}_2^{j_2}, \ldots, \mathcal{C}_\alpha^{j_\alpha}\}$ and for each set $C \in \mathcal{C}$, we add a set $P \in \mathcal{P}$ and we let $P = C$. In other words, all vertices in C are added to P. Observe that $|\mathcal{C}| \leq k$. Since each $\mathbf{C}_i$ is of size $2^{\mathcal{O}(k)}$, it is easy to verify that the number of instances is in fact $2^{\mathcal{O}(\alpha k)}$. We now prove the correctness of the algorithm.

Assume that $(G = (V, \cup_{i=1}^{\alpha} E_i), k)$ is a yes-instance and let S be a solution of size at most k. Note that S need not be a minimal fvs in G_i, $2 \leq i \leq \alpha$. However, for each $i \in \{2, \ldots, \alpha\}$, there exists a set $S' \subseteq S$ such that S' is a minimal fvs for G_i. Hence, by Definition 2.2 and Lemma 2.3, for every $i \in \{2, \ldots, \alpha\}$, there exists a $\mathcal{C}_i^j \in \mathbf{C}_i$ such that for all $C \in \mathcal{C}_i^j$ we have $S' \cap C \neq \emptyset$. Since we enumerate all compact representations and create one instance for each, we know that at least one instance $(G', \mathcal{P}, k', \ell)$ of Colorful OCT will correspond to the correct choice. The fact that S is a solution for $(G', \mathcal{P}, k', \ell)$ follows from the fact that S contains a minimal oct for G_1.

For the other direction, let S' be a solution for an instance $(G', \mathcal{P}, k', \ell)$ of Colorful OCT. Since S' is of size at most k, it is clearly an oct for G_1. Moreover, since S' must intersect every $P \in \mathcal{P}$, it follows from the definition of compact representations and our construction that S' is an fvs for G_i, $2 \leq i \leq \alpha$, as needed. ◀

We now focus on solving an instance $(G, \mathcal{P}, k, \ell)$ of Colorful OCT. Recall that we also have access to the set O which is an oct of G of size at most k. Our next step is to reduce $(G, \mathcal{P}, k, \ell)$ to an instance $(G, \mathcal{P}, O, k, \ell)$ of Colorful OCT Compression. The correctness of this reduction is immediate. The final piece in our sequence of reductions is to reduce $(G, \mathcal{P}, O, k, \ell)$ to $2^{\mathcal{O}(k)}$ instances of Colorful Separator. Before we state our final reduction, we need the following.

▶ **Definition 3.2.** Let G be a graph, let O be an oct of G, let $X \subseteq O$, let $\mathcal{Q} = (L, R)$ be a valid bipartition of $G - O$, and let $\mathcal{W} = (A, B)$ be a partition of X. We define the graph $G_{\mathcal{Q},\mathcal{W}}^X$ as the graph obtained from G as follows. Add two new vertices s and t, make s adjacent to all vertices in $(N(A) \cap L) \cup (N(B) \cap R)$, and make t adjacent to all vertices in $(N(A) \cap R) \cup (N(B) \cap L)$. Finally, delete X.

▶ **Proposition 1.** [7] *Let G be a graph, let X be an oct of G, and let $\mathcal{Q}$ be a valid bipartition of $G - X$. Then the following statements hold.*

(i) *Let Y be an oct of G, $Z = X \setminus Y$, let $G' = G - (X \cap Y)$, and let $\mathcal{Q}' = \mathcal{Q}\big|_{V(G')}$. Then, there is a partition $\mathcal{Z} = (Z_1, Z_2)$ of Z such that $Y \setminus X$ is an (s,t)-separator in $G'^{Z}_{\mathcal{Q}',\mathcal{Z}}$.*

(ii) *Let $Z \subseteq X$, let $G' = G - Z$, and let $\mathcal{Q}' = \mathcal{Q}\big|_{V(G')}$. Let $\mathcal{B}$ be a valid bipartition of $X \setminus Z$. If Y is an (s,t)-separator in the graph $G'^{X\setminus Z}_{\mathcal{Q}',\mathcal{B}}$, then $Y \cup Z$ is an oct of G.*

▶ **Lemma 3.3.** *There is an algorithm that, given an instance $(G, \mathcal{P}, O, k, \ell)$ of* Colorful OCT Compression*, runs in time $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ and returns a set of $2^{\mathcal{O}(k)}$ instances of* Colorful Separator *such that the original instance is a yes-instance if and only if at least one of the returned instances is a yes-instance.*

Proof. First, we let $\mathcal{Q}$ be an arbitrary valid bipartition of $G - O$. Next, for each of the at most 3^k partitions of O into three sets X, Y, and Z, we make sure that the following conditions hold. We check that $O = X \cup Y \cup Z$, $G[X]$ and $G[Y]$ are edgeless ((X, Y) is a valid bipartition of $O \setminus Z$), and $P \not\subseteq X \cup Y$, for any $P \in \mathcal{P}$. If all conditions hold, we construct an instance $(G'^{X\cup Y}_{\mathcal{Q}',\mathcal{W}}, \mathcal{P}', s, t, k', \ell')$ of Colorful Separator where $G' = G - Z$, $\mathcal{Q}'$ is the bipartition of $G - O$ restricted to $V(G')$, $\mathcal{W}$ is the partition (X, Y), and the graph $G'^{X\cup Y}_{\mathcal{Q},\mathcal{W}}$ is obtained from G' by adding vertices s and t and deleting $X \cup Y$ (see Definition 3.2). We set $k' = k - |Z|$. It remains to show how to construct $\mathcal{P}' = \{P'_1, \ldots, P'_{\ell'}\}$ from $\mathcal{P} = \{P_1, \ldots, P_\ell\}$. Let $\mathcal{P}''$ be the subset of $\mathcal{P}$ consisting of all sets $P \in \mathcal{P}$ such that $P \cap Z \neq \emptyset$. For every $P \in \mathcal{P} \setminus \mathcal{P}''$, we assign a unique index $j \in [|\mathcal{P} \setminus \mathcal{P}''|]$ and set $P'_j = P$. It follows that $\ell' \leq \ell$.

We now claim that the given instance $(G, \mathcal{P}, O, k, \ell)$ is a yes-instance of Colorful OCT Compression if and only if at least one of the constructed 3^k instances $(G'^{X\cup Y}_{\mathcal{Q}',\mathcal{W}}, \mathcal{P}', s, t, k - |Z|, \ell')$ is a yes-instance of Colorful Separator. We first argue the forward direction. Suppose that $(G, \mathcal{P}, O, k, \ell)$ is a yes-instance of Colorful OCT Compression. Let $S \subseteq V(G)$ denote a set of vertices of size at most k such that $G - S$ is bipartite and $S \cap P \neq \emptyset$, for each $P \in \mathcal{P}$. Let $Z = S \cap O$ and $\mathcal{P}'' = \{P \in \mathcal{P} \mid P \cap Z \neq \emptyset\}$. Then, from statement (i) of Proposition 1, there is a partition $\mathcal{R}$ of $O \setminus S$ into sets X and Y such that $S \setminus O$ is an (s,t)-separator in the graph $G'^{O\setminus S}_{\mathcal{Q}',\mathcal{R}}$, where $G' = G - Z$. Moreover, for each $P \in \mathcal{P} \setminus \mathcal{P}''$, $(S \setminus O) \cap P \neq \emptyset$. Therefore, $(G'^{O\setminus S}_{\mathcal{Q}',\mathcal{R}}, \mathcal{P}', s, t, k - |Z|, \ell')$ is a yes-instance of Colorful Separator.

For the reverse direction, suppose that there is a partition of O into sets X, Y, and Z such that $(G'^{X\cup Y}_{\mathcal{Q}',\mathcal{W}}, \mathcal{P}', s, t, k - |Z|, \ell')$ is a yes-instance of Colorful Separator. Then, let Y be a corresponding solution. That is, Y is an (s,t)-separator in $G'^{X\cup Y}_{\mathcal{Q}',\mathcal{W}}$ of size at most $k - |Z|$, where $G' = G - Z$. By the second statement of Proposition 1, $Y \cup Z$ is an oct of G of size at most k. Moreover, for each $P \in \mathcal{P}$, either $Z \cap P \neq \emptyset$ or $Y \cap P \neq \emptyset$. Hence, $(G, \mathcal{P}, O, k, \ell)$ is a yes-instance of Colorful OCT Compression. ◀

To summarize, given an instance $(G = (V, \cup_{i=1}^{\alpha} E_i), k)$ of Simultaneous FVS/OCT, we first compute an odd cycle transversal of G_1 and a feedback vertex set of G_i, $i \in [\alpha] \setminus \{1\}$, in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time. Then, we generate $2^{\mathcal{O}(\alpha k)}$ instances of Colorful OCT, of the form $(G, \mathcal{P}, k, \ell \leq \alpha k)$, in $\mathcal{O}^\star(2^{\mathcal{O}(\alpha k)})$ time. Each instance of Colorful OCT is converted into an instance $(G, \mathcal{P}, O, k, \ell)$ of Colorful OCT Compression in polynomial time. Finally, for each instance of Colorful OCT Compression we generate $2^{\mathcal{O}(k)}$ instances of Colorful Separator, with parameters k and $\ell \leq \alpha k$, in $\mathcal{O}^\star(2^{\mathcal{O}(k)})$ time. Lemmas 3.1 and 3.3 together imply that if we can solve an instance of Colorful Separator in $\mathcal{O}^\star(k^{\mathsf{poly}(\alpha,k)})$ time then the algorithm for Simultaneous FVS/OCT follows. We describe such an algorithm in the next section.

4 An FPT algorithm for finding colorful separators

We in fact give an algorithm for a more general problem, which we call Colorful Multiway Cut (or CMWC for short). Before we proceed, we need a few definitions.

▶ **Definition 4.1.** Given a graph G, a set $T \subseteq V(G)$, and a partition $\mathcal{T}$ of T into (pairwise disjoint) sets $\{T_1, \ldots, T_r\}$, we say that $S \subseteq V(G) \setminus T$ is a $\mathcal{T}$*-multiway cut* if, in $G - S$, no vertex in $T_i \setminus S$ can reach a vertex in $T_j \setminus S$, for all $i, j \in [r]$, such that $i \neq j$. We say that $\mathcal{T}$ is an *edge-free partition* of T if there are no edges (u, v) in $G[T]$ where u and v belong to different sets of $\mathcal{T}$.

Given a grouping $\{P_1, \ldots, P_\ell\}$ of the vertices of a graph G, we define a partial coloring function $\mathsf{col} : V(G) \to 2^{[\ell]}$. That is, we have $i \in \mathsf{col}(v)$ if and only if $v \in P_i$, for some $i \in [\ell]$. In this context, for a set $C \subseteq [\ell]$, a subset S of vertices of G is called C*-colorful* if, for each $i \in C$, there is a vertex v in S such that $i \in \mathsf{col}(v)$. For a subset $S \subseteq V(G)$, we denote by $\mathsf{col}(S)$ the set $\{j \mid v \in S \cap (\bigcup_{i=1}^{\ell} P_i) \wedge j \in \mathsf{col}(v)\}$, i.e. the set of colors appearing in S. The CMWC can now be defined as follows.

Colorful Multiway Cut (CMWC)	**Parameter(s):** k, $\lvert T\rvert$, and ℓ

Input: A graph $G = (V, E)$, a set $T \subseteq V(G)$, a partition $\mathcal{T}$ of T into (pairwise disjoint) sets $\{T_1, \ldots, T_r\}$, a grouping $\mathcal{P}$ of the vertices of G into (not necessarily distinct) sets $\{P_1, \ldots, P_\ell\}$, a set $C \subseteq [\ell]$, and an integer k.
Question: Is there a set $S \subseteq V(G) \setminus T$ such that $|S| \leq k$, S is a $\mathcal{T}$-multiway cut in G, and S is C-colorful?

4.1 Setting up the algorithm

Let $(G, T, \mathcal{T}, \mathcal{P}, C, k)$ be an instance of CMWC. We start by stating a few simple reduction rules (which are applied in the order they are stated).

▶ Reduction Rule 1. If $k < 0$ then return false, i.e. $(G, T, \mathcal{T}, \mathcal{P}, C, k)$ is a no-instance.

▶ Reduction Rule 2. If $k = 0$ and $\emptyset$ is a solution to $(G, T, \mathcal{T}, \mathcal{P}, C, k)$ then return true, i.e. $(G, T, \mathcal{T}, \mathcal{P}, C, k)$ is yes-instance. If $k = 0$ and $\emptyset$ is not a solution then return false.

▶ Reduction Rule 3. If there exists $i \in C$ such that $P_i \subseteq T$ then return false.

▶ Reduction Rule 4. If there exists $i \in C$ such that $P_i \cap T \neq \emptyset$ then set $P_i = P_i \setminus T$.

▶ Reduction Rule 5. If there exists $i \in C$ such that $P_i = \emptyset$ then return false.

▶ Reduction Rule 6. If $\mathcal{T}$ is not an edge-free partition then return false.

It is easy to see that Reduction Rules 1 to 6 are safe and can be applied in polynomial time. When $k > 0$ and $\emptyset$ is a $\mathcal{T}$-multiway cut, we can solve the corresponding instance in time $\mathcal{O}^\star(2^{\mathcal{O}(\ell)})$. The following observation describes how.

▶ Observation 1. Let $\mathcal{I} = (G, T, \mathcal{T}, \mathcal{P}, C, k)$ be an instance of Colorful Multiway Cut. If $k > 0$ and $\emptyset$ is a $\mathcal{T}$-multiway cut then $\mathcal{I}$ can be solved in $\mathcal{O}(2^{\mathcal{O}(\ell)} n^2)$ time, where $n = |V(G)|$.

Proof. If $k > 0$ and $\emptyset$ is a $\mathcal{T}$-multiway cut then we are left with the problem of finding a set $S \subseteq V(G) \setminus T$ of size at most k such that $S \cap P_i \neq \emptyset$, for each $i \in C$. Hence, we construct a family $\mathcal{F}$ consisting of a set $f_{P_i} = P_i$ for each for each $i \in C$ and we let $\mathcal{U} = \cup_{i \in C} P_i$. Note that $|\mathcal{F}| \leq \ell \leq \alpha k$ and $|\mathcal{U}| \leq |V(G)|$. Since Reduction Rules 3, 4, and 5 are not applicable, for each $i \in C$, we have $f_{P_i} \neq \emptyset$ and $f_{P_i} \cap T = \emptyset$. If we can find a subset $U \subseteq \mathcal{U}$ which intersects all the sets in $\mathcal{F}$, such that $|U| \leq k$, then U is the required solution. Otherwise, we

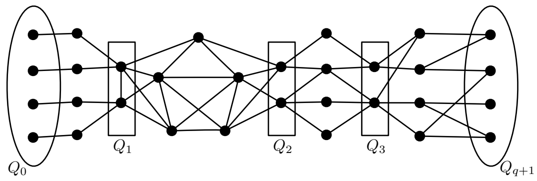

Figure 1 An illustration of a tight separator sequence.

have a no-instance. It is known that the HITTING SET problem parameterized by the size of the family $\mathcal{F}$ is fixed-parameter tractable and can be solved in $\mathcal{O}(2^{\mathcal{O}(|\mathcal{F}|)}|\mathcal{U}|^2)$ time [7]. In particular, we can find an optimum hitting set $U \subseteq \mathcal{U}$, hitting all the sets in $\mathcal{F}$. Therefore, we have a subset of vertices that intersects all sets P_i, for $i \in C$. ◀

Before proceeding with the description of the algorithm, we first recall the notion of tight separator sequences introduced in [19]. However, the definition and structural lemmas regarding tight separator sequences used in this paper are from [20]. Note that although [20] contains Definition 4.2 and Lemma 4.3 in terms of directed graphs, the same holds true for undirected graphs because one can represent any undirected graph as a directed graph by adding bidirectional edges between every pair of adjacent vertices.

▶ **Definition 4.2.** Let X and Y be two subsets of $V(G)$ and let $k \in \mathbb{N}$. A *tight (X,Y)-reachability sequence* of order k is an ordered collection $\mathcal{H} = \{H_0, H_1, \ldots, H_q, H_{q+1}\}$ of sets in $V(G)$ satisfying the following properties:
- $X \subseteq H_i \subseteq V(G) \setminus N[Y]$ for any $0 \leq i \leq q$;
- $X = H_0 \subset H_1 \subset H_2 \subset \ldots \subset H_q \subset H_{q+1} = V(G) \setminus Y$;
- H_i is reachable from X in $G[H_i]$ and every vertex in $N(H_i)$ can reach Y in $G - H_i$ (implying that $N(H_i)$ is a minimal (X,Y)-separator in G);
- $|N(H_i)| \leq k$ for every $1 \leq i \leq q$;
- $N(H_i) \cap N(H_j) = \emptyset$ for all $1 \leq i, j \leq q$ and $i \neq j$;
- For any $0 \leq i \leq q-1$, there is no (X,Y)-separator S of size at most k where $S \subseteq H_{i+1} \setminus N[H_i]$ or $S \cap N[H_q] = \emptyset$ or $S \subseteq H_1$.

We let $Q_0 = X$, $Q_i = N(H_i)$, for $1 \leq i \leq q$, $Q_{q+1} = Y$, and $\mathcal{Q} = \{Q_0, Q_1, \ldots, Q_q, Q_{q+1}\}$. We call $\mathcal{Q}$ a *tight (X,Y)-separator sequence* of order k.

▶ **Lemma 4.3.** *(see [20]) There is an algorithm that, given a graph G on n vertices and m edges, subsets $X, Y \subseteq V(G)$ and $k \in \mathbb{N}$, runs in time $\mathcal{O}(k^2nm)$ and either correctly concludes that there is no (X,Y)-separator of size at most k in G or returns the sets $H_0, H_1, H_2 \setminus H_1, \ldots, H_q \setminus H_{q-1}, H_{q+1} \setminus H_q$ corresponding to a tight (X,Y)-reachability sequence $\mathcal{H} = \{H_0, H_1, \ldots, H_q, H_{q+1}\}$ of order k.*

See Figure 1 for an illustration of a tight (X,Y)-separator sequence. Our algorithm will be a combination of dynamic programming over the sets Q_i, $0 \leq i \leq q+1$, and recursive calls for solving "smaller" instances of the same problem. Below we state some observations that help understand the structure of a solution and are crucial for achieving the stated running time.

▶ Observation 2. Let $(G, T, \mathcal{T}, \mathcal{P}, C, k)$ be an instance of COLORFUL MULTIWAY CUT and let T_1 be a set in $\mathcal{T}$ which is linked to some set in $\mathcal{T} \setminus \{T_1\}$. Moreover, let $\mathcal{H} = \{H_0, H_1, \ldots, H_q, H_{q+1}\}$ be a tight $(T_1, T \setminus T_1)$-reachability sequence of order k and let $\mathcal{Q} = \{Q_0, Q_1, \ldots, Q_q, Q_{q+1}\}$ be the corresponding tight separator sequence. Assume $(G, T, \mathcal{T},$

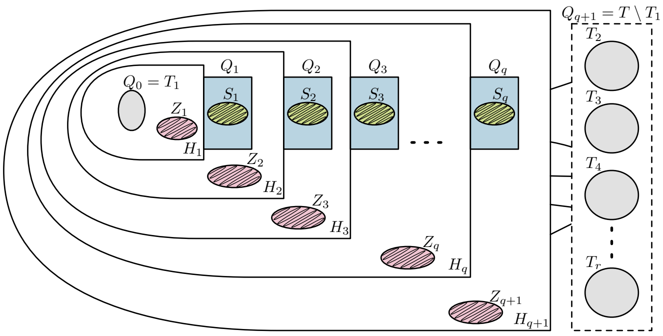

Figure 2 An illustration of the division of a solution S into various sets.

$\mathcal{P}$, C, k) is a yes-instance and let S be one of its solution. Then, S can be partitioned into the following (pairwise-disjoint) sets (see Figure 2).

- $Z_1 = S \cap (H_1 \setminus Q_0)$.
- $S_i = S \cap Q_i$ for $1 \leq i \leq q$.
- $Z_i = (S \cap (H_i \setminus N[H_{i-1}])) \setminus Q_{q+1}$ for $2 \leq i \leq q+1$.

We invoke the last property of tight separator sequences to obtain the following bound.

▶ Observation 3. $|Z_i| \leq k-1$ for each $i \in [q+1]$.

Observation 3 is crucial as it allows us to apply our algorithm on sub-instances with a strictly smaller parameter.

Algorithm 1: Pseudocode for **ALG1**

Input: $(G, T, \mathcal{T}, \mathcal{P}, C, k)$
Output: true or false

1 Apply all reduction rules (in order) and return true/false appropriately (if applicable).
2 **if** $k > 0$ *and* $\emptyset$ *is a* $\mathcal{T}$*-multiway cut* **then**
3   **return** true/false appropriately (Observation 1)
4 Let $T_1 \in \mathcal{T}$ such that T_1 is linked to some $T_j \in \mathcal{T}$, where $j \neq 1$.
5 Let $\mathcal{H} = \{H_0, H_1, \ldots, H_q, H_{q+1}\}$ be a $(T_1, T \setminus T_1)$-reachability sequence of order k;
6 Let $\mathcal{Q} = \{Q_0, Q_1, \ldots, Q_q, Q_{q+1}\}$ be the corresponding $(T_1, T \setminus T_1)$-separator sequence;
7 **if** $\mathcal{Q} = \emptyset$ **then**
8   **return** false;
9 **return** **ALG2**$(G, T, \mathcal{T}, \mathcal{P}, C, k, \mathcal{Q})$;

To keep the presentation clean, we shall define two routines **ALG1** and **ALG2**. **ALG1** (Algorithm 1) delegates most of the "heavy lifting" to **ALG2**. That is, **ALG1** simply checks if any of the reduction rules are applicable and solves the instance if it corresponds to one of the base cases. When this is not the case, **ALG1** proceeds by computing a tight separator sequence and calls **ALG2**. Note that we can safely return false when the algorithm fails to construct such a sequence (Lines 7 and 8 of Algorithm 1). We now move to the description of **ALG2**, which takes as additional input the newly constructed tight separator sequence. Roughly speaking, **ALG2** will recursively solve a "large" number of instances restricted to graphs that "reside" between two consecutive separators of a separator sequence. The number of instances will be bounded by the number of possible "interactions" between the

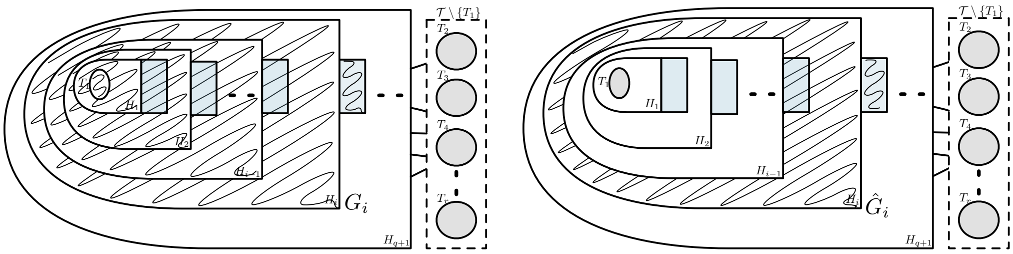

Figure 3 An illustration of graphs in Definition 4.4.

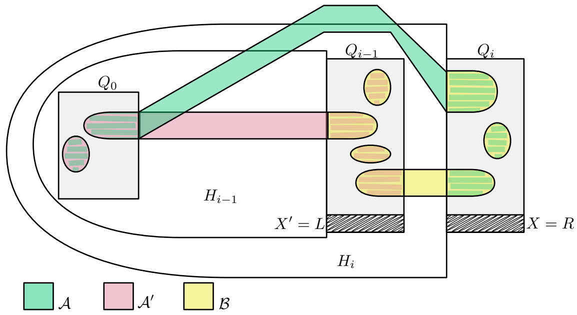

Figure 4 An illustration of compatible tuples.

two consecutive separators and a hypothetical solution. However, due to Observation 3, each one of those recursive calls can be made with a strictly smaller value of k. Having solved all such instances (and stored the outcomes in tables), **ALG2** then proceeds using a dynamic programming routine which computes the answer in a left-to-right manner, i.e. starting from Q_0 all the way to Q_{q+1}. We now give a formal description.

▶ **Definition 4.4.** For a graph G and a tight separator sequence $\mathcal{Q} = \{Q_0, Q_1, \ldots, Q_q, Q_{q+1}\}$, we let $G_i = G - R_G(Q_{q+1}, Q_i)$, *i.e.* the graph obtained after removing the vertices that are reachable from Q_{q+1} after deleting Q_i, and we let $\widehat{G}_i = G_i - (V(G_{i-1}) \setminus Q_{i-1})$ (see Figure 3).

For each graph G_i, $i \in [q+1]$, we maintain a table Γ_i, where each entry is indexed by a tuple $(X, \mathcal{A}, \overline{C}, \overline{p})$. For each graph $\widehat{G}_i$, $i \in [q+1]$, we maintain a table Λ_i, where each entry is indexed by a tuple $(L, R, \mathcal{B}, \widehat{C}, \widehat{p})$. The tuples are described below.

- $X \subseteq Q_i \setminus T$ and $L \subseteq Q_{i-1} \setminus T$ and $R \subseteq Q_i \setminus T$;
- $\mathcal{A}$ is an edge-free partition of $(Q_i \cup Q_0) \setminus X$;
- $\mathcal{B}$ is an edge-free partition of $(Q_{i-1} \cup Q_i) \setminus (L \cup R)$;
- $\overline{C}, \widehat{C} \subseteq [\ell]$ and $\overline{p} \leq k - |X|$ and $\widehat{p} \leq k - |L \cup R|$ if $L \cup R \neq \emptyset$ and $\widehat{p} \leq k - 1$, otherwise.

▶ **Definition 4.5.** For a tuple $\tau = (X, \mathcal{A}, \overline{C}, \overline{p})$, we denote by $\mathbb{I}_\tau$ the instance $(G_i - X, (Q_i \cup Q_0) \setminus X, \mathcal{A}, \mathcal{P}|_{V(G_i - X)}, \overline{C}, \overline{p})$ of CMWC. Similarly, for a tuple $\tau = (L, R, \mathcal{B}, \widehat{C}, \widehat{p})$, we denote by $\mathbb{I}_\tau$ the instance $(\widehat{G}_i - (L \cup R), (Q_{i-1} \cup Q_i) \setminus (L \cup R), \mathcal{B}, \mathcal{P}|_{V(G_i - (L \cup R))}, \widehat{C}, \widehat{p})$ of CMWC. Finally, we define $\Gamma_i(\tau)$ (or $\Lambda_i(\tau)$)= `true` if and only if $\mathbb{I}_\tau$ is a yes-instance of CMWC.

▶ **Definition 4.6.** Given three tuples $\tau_1 = (X, \mathcal{A}, \overline{C}, \overline{p})$, $\tau_2 = (L, R, \mathcal{B}, \widehat{C}, \widehat{p})$, and $\tau_3 = (X', \mathcal{A}', \overline{C}', \overline{p}')$, we say that they are *compatible* if all of the following conditions hold (see Figure 4).

- $\tau_1 \in \Gamma_i$ and $\tau_2 \in \Lambda_i$ and $\tau_3 \in \Gamma_{i-1}$, where $i \in [q+1]$;
- $X' = L$ and $X = R$;
- $\mathcal{A}\big|_{Q_i \setminus X} = \mathcal{B}\big|_{Q_i \setminus R}$ and $\mathcal{B}\big|_{Q_{i-1} \setminus L} = \mathcal{A}'\big|_{Q_{i-1} \setminus X'}$ and $\mathcal{A}\big|_{Q_0} = \mathcal{A}'\big|_{Q_0}$;
- $\overline{p}' + \widehat{p} + |L| \leq p$ and $\overline{C}' \cup \widehat{C} \cup \mathsf{col}(L) = \overline{C}$.

Algorithm 2: Pseudocode for **ALG2**

Input: $(G, T, \mathcal{T}, \mathcal{P}, C, k, \mathcal{Q})$

Output: true or false

1 Initialize all entries in Γ_i to false, for $i \in [q+1]$;

2 Initialize all entries in Λ_i to false, for $i \in [q+1]$;

3 **for** *each* $\widehat{G}_i \in \{\widehat{G}_1, \ldots, \widehat{G}_{q+1}\}$ **do**

4   **for** *each* $L \subseteq Q_{i-1} \setminus T$ *and each each* $R \subseteq Q_i \setminus T$ **do**

5   **for** *each edge-free partition* $\mathcal{B}$ *of* $(Q_{i-1} \cup Q_i) \setminus (L \cup R)$ **do**

6   **for** *each* $\widehat{C} \subseteq [\ell]$ *and each* $0 \leq \widehat{p} \leq k - \max\{1, |L \cup R|\}$ **do**

7                 $\mathbb{I} = (\widehat{G}_i - (L \cup R), (Q_{i-1} \cup Q_i) \setminus (L \cup R), \mathcal{B}, \mathcal{P}\big|_{V(G_i - (L \cup R))}, \widehat{p})$;

8                 $\Lambda_i(L, R, \mathcal{B}, \widehat{C}, \widehat{p}) =$ **ALG1**$(\mathbb{I})$;

9 Copy table entries for Γ_1, *i.e.* $\Gamma_1(X, \mathcal{A}, \overline{C}, \overline{p}) = \Lambda_1(\emptyset, X, \mathcal{A}, \overline{C}, \overline{p})$;

10 **for** *each* $G_i \in \{G_2, \ldots, G_{q+1}\}$ *(in order)* **do**

11   **for** *each* $X \subseteq Q_i \setminus T$ **do**

12   **for** *each edge-free partition* $\mathcal{A}$ *of* $(Q_i \cup Q_0) \setminus X$ **do**

13   **for** *each* $\overline{C} \subseteq [\ell]$ *and each* $0 \leq \overline{p} \leq k - |X|$ **do**

14                 $\tau_1 = (X, \mathcal{A}, \overline{C}, \overline{p})$;

15   **for** *each tuple* $\tau_2 = (L, R, \mathcal{B}, \widehat{C}, \widehat{p}) \in \Lambda_i$ **do**

16   **for** *each tuple* $\tau_3 = (X', \mathcal{A}', \overline{C}', \overline{p}') \in \Gamma_{i-1}$ **do**

17   **if** τ_1, τ_2, *and* τ_3 *are compatible* **then**

18                             $\Gamma_i(\tau_1) = \Gamma_i(\tau_1) \vee [\Gamma_{i-1}(\tau_3) \wedge \Lambda_i(\tau_2)]$;

19 **if** $\Gamma_{q+1}(\emptyset, \mathcal{T}, C, p) =$ *true (for some* $p \leq k$*)* **then**

20   **return** true;

21 **return** false;

The complete description of **ALG2** is given in Algorithm 2. Initially, we set all table entries to false (Lines 1 and 2). Then, for each $\widehat{G}_i \in \{\widehat{G}_1, \ldots, \widehat{G}_{q+1}\}$ and for each possible tuple $(L, R, \mathcal{B}, \widehat{C}, \widehat{p}) \in \Lambda_i$, we solve the corresponding CMWC instance $\mathcal{I} = (\widehat{G}_i - (L \cup R), (Q_{i-1} \cup Q_i) \setminus (L \cup R), \mathcal{B}, \mathcal{P}\big|_{V(G_i - (L \cup R))}, \widehat{p})$. That is, we set $\Lambda_i(L, R, \mathcal{B}, \widehat{C}, \widehat{p})$ if $\mathcal{I}$ is a yes-instance (Lines 3 to 8). Having computed all those values, we then proceed to filling table Γ_1. Since G_0 is a subgraph of G_1, and $G_1 = \widehat{G}_1$, we simply set $\Gamma_1(X, \mathcal{A}, \overline{C}, \overline{p}) = \Lambda_1(\emptyset, X, \mathcal{A}, \overline{C}, \overline{p})$ (for all tuples). This is justified by the fact that a solution is not allowed to delete any vertex in Q_0. To complete table Γ_i, $i > 1$, we simply use the following:

$$\Gamma_i(X, \mathcal{A}, \overline{C}, \overline{p}) = \bigvee [\Gamma_{i-1}(X', \mathcal{A}', \overline{C}', \overline{p}') \wedge \Lambda_i(L, R, \mathcal{B}, \widehat{C}, \widehat{p})],$$

where tuples $(X, \mathcal{A}, \overline{C}, \overline{p})$, $(X', \mathcal{A}', \overline{C}', \overline{p}')$, and $(L, R, \mathcal{B}, \widehat{C}, \widehat{p})$ are compatible. Finally, **ALG2** returns true whenever there exists a tuple $\Gamma_{q+1}(\emptyset, \mathcal{T}, C, p) =$ true (for some $p \leq k$).

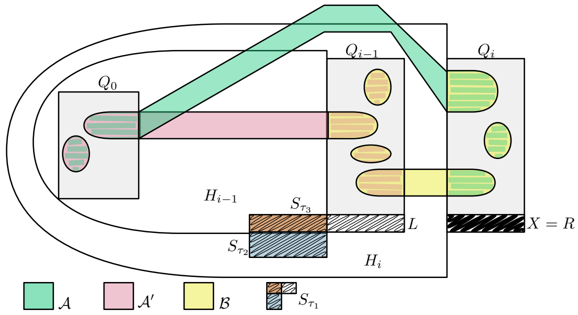

Figure 5 An illustration of the proof of Lemma 4.7.

4.2 Correctness and runtime analysis

We are now ready to prove our main structural lemma which reduces the computation of the entries in Γ_i (when $i > 1$) to those in Γ_{i-1} and Λ_i. The lemma is proved in a purely existential setting and serves as the proof of correctness of the algorithm.

▶ **Lemma 4.7.** *For any $i \in [q+1]$ and tuple $\tau_1 = (X, \mathcal{A}, C_1, p_1) \in \Gamma_i$, $\mathbb{I}_{\tau_1}$ is a yes-instance if and only if there is a tuple $\tau_2 = (L, R, \mathcal{B}, C_2, p_2) \in \Lambda_i$ and a tuple $\tau_3 = (X', \mathcal{A}', C_3, p_3) \in \Gamma_{i-1}$ such that $\mathbb{I}_{\tau_2}$ and $\mathbb{I}_{\tau_3}$ are both yes-instances and all three tuples are compatible.*

Proof. We begin with the forward direction. Suppose that $\mathbb{I}_{\tau_1}$ is a yes-instance. Then, it must be the case that there is a C_1-colorful set $S_{\tau_1} \subseteq V(G_i) \setminus X$ of size at most p_1 such that S_{τ_1} is an $\mathcal{A}$-multiway cut in G_i. We define sets $S_{\tau_2} = S_{\tau_1} \cap (V(\widehat{G_i}) \setminus (Q_{i-1} \cup Q_i))$ and $S_{\tau_3} = S_{\tau_1} \cap (V(G_{i-1}) \setminus Q_{i-1})$ (see Figure 5). We will now define tuples τ_2 and τ_3. Let $C_2 = \mathsf{col}(S_{\tau_2})$, $C_3 = \mathsf{col}(S_{\tau_3})$, $p_2 = |S_{\tau_2}|$, $p_3 = |S_{\tau_3}|$, $X' = L = S_{\tau_1} \cap Q_{i-1}$ and $R = X$. Furthermore, we let $\mathcal{C}$ be the equivalence class defined on the vertex sets of connected components in $G_i - (S_{\tau_1} \cup X)$. We set $\mathcal{B} = \mathcal{C}|_{(Q_{i-1} \cup Q_i) \setminus L}$ and $\mathcal{A}' = \mathcal{C}|_{Q_0 \cup Q_{i-1}}$. Finally, we set $\tau_2 = (L, R, \mathcal{B}, C_2, p_2)$ and $\tau_3 = (X', \mathcal{A}', C_3, p_3)$. We will now argue that S_{τ_2} and S_{τ_3} are solutions for $\mathbb{I}_{\tau_2}$ and $\mathbb{I}_{\tau_3}$, respectively. For two distinct sets $B, B' \in \mathcal{B}$ we have no path between them in $G_{\tau_2} - S_{\tau_2}$ since they belong to different connected components in $G_i - (S_{\tau_1} \cup X)$, $\widehat{G_i}$ is a subgraph of G_i, Q_{i-1} is a (Q_0, Q_i)-separator, and by the definition of $\mathcal{B}$. This proves that S_{τ_2} is a solution for $\mathbb{I}_{\tau_2}$. An analogous argument can be given for S_{τ_3} being a solution for $\mathbb{I}_{\tau_3}$. This completes the argument in the forward direction.

In the reverse direction, suppose that there are $\tau_2 = (L, R, \mathcal{B}, C_2, p_2) \in \Lambda_i$ and $\tau_3 = (X', \mathcal{A}', C_3, p_3) \in \Gamma_{i-1}$ compatible with $\tau_1 \in \Gamma_i$ such that $\mathbb{I}_{\tau_3}$ and $\mathbb{I}_{\tau_2}$ are yes-instances. Let S_{τ_2} and S_{τ_3} be solutions for the respective instances.

We claim that $S_{\tau_1} = S_{\tau_2} \cup S_{\tau_3} \cup L$ is a solution for $\mathbb{I}_{\tau_1}$. Notice that $p_1 \leq p_2 + p_3 + |L|$ and $C_1 = C_2 \cup C_3 \cup \mathsf{col}(L)$ (we have $R = X$). We need to show that S_{τ_1} is an $\mathcal{A}$-multiway cut in $G_i - X$. Targeting a contradiction, suppose there are distinct sets $A_1, A_2 \in \mathcal{A}$ such that A_1 and A_2 are linked. Then, there exists a path P from $a_1 \in A_1$ to a vertex $a_2 \in A_2$. Consider the following ordered sequence of vertices $x_1, x_2, \ldots, x_t$ in $V(P) \cap (Q_0 \cup Q_{i-1} \cup Q_i)$ obtained from P, i.e. by the order in which they appear in P. Notice that each of the subpaths P_j from (x_j, x_{j+1}) of P, for $j \in [t-1]$, is completely contained in one of $\widehat{G_i} - S_{\tau_2}$ or $G_{i-1} - S_{\tau_3}$. But this implies that there exists $B \in \mathcal{B}$ such that $\{x_j \mid j \in [t]\} \cap (Q_{i-1} \cup Q_i) \subseteq B$. Similarly, there exists $A' \in \mathcal{A}'$ such that $\{x_j \mid j \in [t]\} \cap (Q_0 \cup Q_{i-1}) \subseteq A'$. But since τ_1, τ_2, and τ_3 are compatible therefore, a_1 and a_2 must belong to the same set in $\mathcal{A}$, contradicting the choice of a_1 and a_2. This concludes the proof. ◀

The above lemma proves the correctness of our algorithm since we set $\Gamma_i(\tau_1)$ to be true precisely when there are τ_2 and τ_3 such that $\Gamma_{i-1}(\tau_2) = \Lambda_i(\tau_3) = \mathsf{true}$. Finally, we prove the claimed running time bound.

▶ **Theorem 4.8.** Colorful Multiway Cut *can be solved in* $\mathcal{O}^\star((k+t)^{\mathcal{O}(kt+k^3)}2^{\mathcal{O}(\ell k)})$ *time, where* $t = |T|$.

Proof. Let $n = |V(G)|$, $m = |E(G)|$, $t = |T|$, and $\mathbb{T}(n,t,\ell,k)$ denote the "local" time taken by our algorithm to solve an instance $(G, T, \mathcal{T}, \mathcal{P}, C, k)$ of Colorful Multiway Cut. By local, we mean the time taken ignoring all recursive calls. At the base case, the algorithm correctly decides the instance in $\mathcal{O}(2^{\mathcal{O}(\ell)}n^2)$ time (Observation 1). Hence, at the base case we have $\mathbb{T}(n,t,\ell,k) = \mathcal{O}(2^{\mathcal{O}(\ell)}n^2)$. Otherwise, we have $\mathbb{T}(n,t,\ell,k) \leq \mathcal{O}(k^2nm) + \mathcal{O}(n(t+k)^{t+k}2^k2^\ell)$, where $\mathcal{O}(k^2nm)$ is the time taken to construct a tight separator sequence (Lemma 4.3) and $\mathcal{O}(n(t+k)^{t+k}2^k2^\ell)$ is the time taken to compute all table entries Γ_i, for $i \in [q+1] \cup \{0\}$. Stated differently, $\mathcal{O}((t+k)^{t+k}2^k2^\ell)$ is the size of the largest table. The correctness of this step follows from Lemma 4.7 and the description of our two subroutines **ALG1** and **ALG2**.

Now consider the recursion tree. We let N_d denote a node in this tree at depth d. Note that the depth of our recursion tree is at most k; since k decreases by at least one in every recursive call (Observation 3 and Definition 4.5). Consider any node N_d in the recursion tree with associated measures (n', t', ℓ', k'), i.e. $N_d = (n', t', \ell', k')$. We have $n' \leq n$, $k' \leq k$, $\ell' \leq \ell$, and $t' \leq \max(2k, t+k^2) \leq t + 2k + k^2$ (since t is either $2k$ or increases by at most k when computing Γ_{q+1} and the depth of our recursion tree is at most k). Moreover, if we sum n' for all nodes at depth d in the recursion tree we get $\sum_{N_d} n' \leq (\sum_{N_{d-1}} n')\mathcal{O}((t+3k+k^2)^{t+3k+k^2}2^k2^\ell)$ (since $t' \leq t+2k+k^2$). Therefore, at the deepest level, i.e. level k, we get:

$$\sum_{N_k} n' \leq n \cdot \mathcal{O}(((t+3k+k^2)^{t+3k+k^2}2^k2^\ell)^k) = n \cdot \mathcal{O}((t+3k+k^2)^{kt+3k^2+k^3}2^{k^2}2^{k\ell}).$$

Replacing for n in $\mathbb{T}(n,t,\ell,k)$, we get:

$$\mathbb{T}(n,t,\ell,k) \leq (k+t)^{\mathcal{O}(kt+k^3)}2^{\mathcal{O}(\ell k)}n^{\mathcal{O}(1)}.$$

Multiplying by the number of nodes in the recursion tree, which is bounded by $\mathcal{O}((k+t)^{\mathcal{O}(kt+k^3)}2^{\mathcal{O}(\ell k)}n^{\mathcal{O}(1)})$, we get the desired running time. ◀

Combining Theorem 4.8 with our series of reductions from Section 3, we have obtain the following corollary (Corollary 4.9).

▶ **Corollary 4.9.** Simultaneous FVS/OCT *can be solved in* $\mathcal{O}^\star(k^{\mathsf{poly}(\alpha,k)})$ *time.*

Proof. Recall that Lemmas 3.1 and 3.3 together imply that if we can solve an instance of Colorful Separator in $\mathcal{O}^\star(k^{\mathsf{poly}(\alpha,k)})$ time then the algorithm for Simultaneous FVS/OCT follows. Any instance of Colorful Separator can be reduced to an instance of Colorful Multiway Cut with $|T| = 2$. From Theorem 4.8, such an instance can be solved in time $\mathcal{O}^\star(k^{\mathcal{O}(k^3)}2^{\mathcal{O}(\alpha k)})$. ◀

5 W[1]-hardness of Simultaneous OCT

In this section we show that Simultaneous OCT is W[1]-hard. For notational convenience, we shall use a different encoding of α-edge-colored graphs. Given a graph G with vertex set

$V(G)$ and edge set $E(G)$, we define a coloring function $\mathsf{col}(e) \subseteq 2^{[\alpha]}$. In particular, when $\alpha = 2$, we have $\mathsf{col}(e) \subseteq 2^{\{1,2\}}$. We start by establishing W[1]-hardness of SIMULTANEOUS CUT. In Section 5.1 we show that SIMULTANEOUS CUT is W[1]-hard, even for $\alpha = 2$, by giving a parameterized reduction from MULTICOLORED CLIQUE. In Section 5.2 we give a parameterized reduction from SIMULTANEOUS CUT to SIMULTANEOUS OCT for the same value of α and hence establish the W[1]-hardness of SIMULTANEOUS OCT for $\alpha = 2$. We note that this also implies W[1]-hardness of SIMULTANEOUS OCT for all $\alpha \geq 2$.

5.1 W[1]-hardness of Simultaneous Cut

The SIMULTANEOUS CUT problem is formally defined below.

SIMULTANEOUS CUT **Parameter(s):** k and α
Input: A graph G, two vertices $s, t \in V(G)$, an integer k, and a coloring function $\mathsf{col} : E(G) \to 2^{[\alpha]}$.
Question: Is there $X \subseteq V(G) \setminus \{s,t\}$ of size at most k such that, for all $i \in [\alpha]$, $G_i - X$ has no (s,t)-paths? Here, for $i \in [\alpha]$, $G_i = (V(G), E_i)$, where $E_i = \{e \in E(G) \mid i \in \mathsf{col}(e)\}$.

We give a parameterized reduction from MULTICOLORED CLIQUE which is known to be W[1]-hard [7]. The MULTICOLORED CLIQUE problem is formally defined below.

MULTICOLORED CLIQUE **Parameter(s):** k
Input: A k-partite graph G with a partition $V_1, V_2, \dots, V_k$ of $V(G)$ such that for all $i, j \in [k]$, $|V_i| = |V_j|$.
Question: Is there $X \subseteq V(G)$ such that, for all $i \in [k]$, $|X \cap V_i| = 1$ and $G[X]$ is a clique?

Given an instance $(G, V_1, V_2, \dots, V_k)$ of MULTICOLORED CLIQUE, we proceed by creating an instance $(G', s, t, k', \mathsf{col}' : E(G') \to 2^{\{1,2\}})$ of SIMULTANEOUS CUT such that (G, V_1, V_2, …, V_k) is a yes-instance of MULTICOLORED CLIQUE if and only if $(G', s, t, k', \mathsf{col}' : E(G') \to 2^{\{1,2\}})$ is a yes-instance of SIMULTANEOUS CUT.

The intuitive description of the parameterized reduction is as follows. Let $(G, V_1, V_2, \dots, V_k)$ be an instance of MULTICOLORED CLIQUE. Since $|V_i| = |V_j|$, for all $i, j \in [k]$, we assume that $|V_i| = |V_j| = n$. Furthermore, we assume that for every $i, j \in [k]$, $i \neq j$, there is at least one edge between V_i and V_j, otherwise, the instance is a trivial no-instance of MULTICOLORED CLIQUE and our reduction will simply output a trivial no-instance of SIMULTANEOUS CUT with $\alpha = 2$. For each $i \in [k]$ we assume an arbitrary (but fixed) ordering on the vertices in V_i. For each $i \in [k]$, we will have a vertex selection gadget $\mathcal{S}_i$ that will be responsible for selecting a vertex in V_i. To achieve this, $\mathcal{S}_i$ will have $k-1$ copies of each vertex in V_i, so that each vertex in V_i has a copy corresponding to every $j \in [k] \setminus \{i\}$. For each $j \in [k] \setminus \{i\}$, we have an (s,t)-path with all edges having color 1. Each path contains exactly one copy of every vertex in V_i. Furthermore, these vertices appear in the order given by the ordering we already fixed on the vertices of V_i (see Figure 6).

The jth copy of the vertex set V_i will be used to ensure that there is an edge between the selected vertex in V_i and a vertex in V_j. The copies of any single vertex will form an (s,t)-separator of size $k-1$. Furthermore, the size of minimum (s,t)-separator in $\mathcal{S}_i$ will be $k-1$ and there will be exactly n distinct minimum separator each of which will correspond to a set comprising of $k-1$ copies of a vertex in V_i. By construction of the gadget and by setting budget constraints appropriately we will ensure that we must select a vertex from each of the $k-1$ copies of V_i, for each $i \in [k]$ and the selected $k-1$ vertices correspond

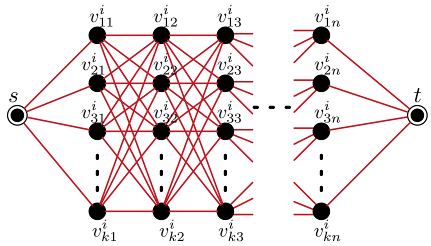

Figure 6 Vertex selection gadget with red color denoting color 1.

to copies of the same vertex, i.e. we select a minimum separator. This will ensure that we have selected exactly one vertex from each V_i, for $i \in [k]$.

For $i, j \in [k]$, $i \neq j$, we will have edge selection gadgets E_{ij} which will ensure that there is an edge selected between V_i and V_j, and the selected edge is incident to the vertex selected from the vertex selection gadget. Finally, we will have a compatibility gadget which will ensure that the edges selected by E_{ij} and E_{ji} correspond to the same edge in G. We need to differentiate between gadgets E_{ij} and E_{ji} for technical reasons that will become clear later. We will now move to the formal description of the reduction.

Construction. Initially, $V(G') = \emptyset$ and $E(G') = \emptyset$. We add two special vertices s and t to $V(G')$, which are the vertices we want to separate, and which will be common to all the gadgets. For $i \in [k]$ we let v^i_j be the jth vertex in V_i. We now formally describe the construction of the various gadgets. We note that the gadgets are not necessarily vertex or edge disjoint (in addition to intersecting with $\{s, t\}$).

Vertex Selection Gadget. For each $i \in [k]$ we have a vertex selection gadget $\mathcal{S}_i$ defined as follows. For each $j \in [k] \setminus \{i\}$, $\mathcal{S}_i$ contains vertices in $V_{ij} = \{v^i_{j1}, v^i_{j2}, \ldots, v^i_{jn}\}$ (refer to Figure 6). Here, the vertices $v^i_{j1}, v^i_{j2}, \ldots, v^i_{jn}$ corresponds to one copy of the vertices $v^i_1, v^i_2, \ldots, v^i_n$ in V_i. Note that for $j, j' \in [k] \setminus \{i\}$ vertices $v^i_{j\ell}, v^i_{j'\ell}$ correspond to copies of the same vertex, namely $v^i_\ell \in V_i$. For $i \in [k]$ and $\ell \in [n]$, we let $V^i_\ell = \{v^i_{j\ell} \mid j \in [k] \setminus \{i\}\}$, i.e. V^i_ℓ denotes the set comprising of $k-1$ copies of the vertex $v^i_\ell \in V_i$. For $i \in [k]$, $\ell \in [n-1]$, and for each $u \in V^i_\ell$ and $u' \in V^i_{\ell+1}$ we add the edge $(u, u') \in E(G')$ and set $\mathsf{col}'((u, u')) = \{1\}$. Note that $G'[V^i_\ell \cup V^i_{\ell+1}]$ is a complete bipartite graph with all edges having the color 1 in their color set. For $i \in [k]$, $u \in V^i_1$ we add the edge $(s, u) \in E(G')$ and set $\mathsf{col}'((s, u)) = \{1\}$. Similarly, for $i \in [k]$, $u \in V^i_n$ we add the edge $(u, t) \in E(G')$ and set $\mathsf{col}'((u, t)) = \{1\}$.

Edge Selection Gadget. For $i \in [k]$ and $j \in [k] \setminus \{i\}$ the edge selection gadget $\mathcal{E}_{ij}$ is constructed as follows. The vertex set of $\mathcal{E}_{ij}$ contains a vertex $e_{uu'}$, for each edge $(u, u') \in E(G)$ with $u \in V_i$ and $u' \in V_j$. We refer the reader to Figure 7 for an illustration. We note here that $\mathcal{E}_{ij}$ and $\mathcal{E}_{ji}$ denote distinct gadgets. For $\ell \in [n]$, we let $E^{ij}_\ell = \{e_{v^i_\ell u'} \mid u' \in V_j, (v^i_\ell, u) \in E(G)\}$, i.e. E^{ij}_ℓ contains vertices corresponding to those edges between V_i and V_j that are incident to the vertex $v^i_\ell \in V_i$. We let $E_{ij} = \cup_{\ell \in [n]} E^{ij}_\ell$. For $\ell \in [n]$ and each $u \in E^{ij}_\ell$, we add the edge $(u, v^i_{j\ell})$ to $\mathcal{E}_{ij}$. We add an induced path P^{ij}_ℓ on the vertices in E^{ij}_ℓ (where the vertices appear in the natural order implied by the ordering of the vertices in V_j) and add these edges to $\mathcal{E}^{ij}_\ell$. For each edge $e \in E(P^{ij}_\ell)$, we let $\mathsf{col}'(e) = \{2\}$. For $\ell \in [n+1]$, we let $\mathbf{K}^{ij}_\ell$ denote a $K_{3,3}$ (complete bipartite graph with 3 vertices on both side) with vertex bipartition $(\{p^{ij}_\ell, q^{ij}_\ell, r^{ij}_\ell\}, \{\bar{p}^{ij}_\ell, \bar{q}^{ij}_\ell, \bar{r}^{ij}_\ell\})$ and add it to $\mathcal{E}_{ij}$. We will refer to $\mathbf{K}^{ij}_\ell$s as *barrier*

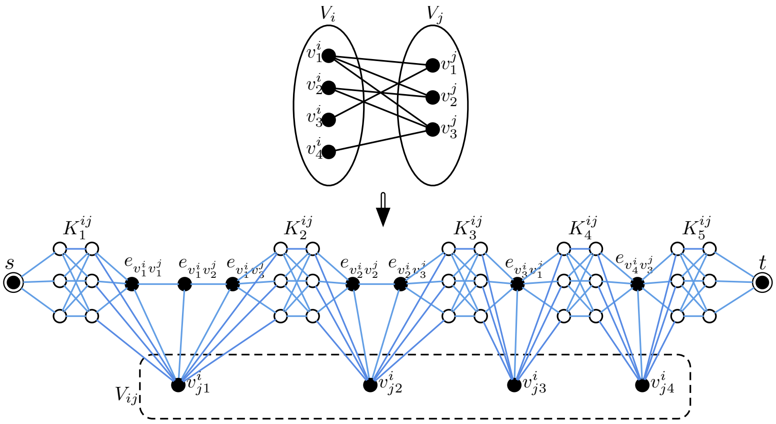

Figure 7 An illustration of an edge selection gadget with blue color denoting color 2.

blocks of $\mathcal{E}_{ij}$. Finally, we join s, t and E^{ij}_ℓ, for $\ell \in [n]$ using the barrier blocks. This is done as follows.

For $\ell \in [n]$, let a^{ij}_ℓ, b^{ij}_ℓ be the first and the last vertex respectively, in the path P^{ij}_ℓ. We add the edges $(a^{ij}_\ell, \bar{p}^{ij}_\ell), (a^{ij}_\ell, \bar{q}^{ij}_\ell), (a^{ij}_\ell, \bar{r}^{ij}_\ell)$ and $(b^{ij}_\ell, p^{ij}_{\ell+1}), (b^{ij}_\ell, q^{ij}_{\ell+1}), (b^{ij}_\ell, r^{ij}_{\ell+1})$ to $E(\mathcal{E}_{ij})$. Also, for $\ell \in [n]$, we add the edges $(v^i_{j\ell}, \bar{p}^{ij}_\ell), (v^i_{j\ell}, \bar{q}^{ij}_\ell), (v^i_{j\ell}, \bar{r}^{ij}_\ell)$ and $(v^i_{j\ell}, p^{ij}_{\ell+1}), (v^i_{j\ell}, q^{ij}_{\ell+1}), (v^i_{j\ell}, r^{ij}_{\ell+1})$ to $E(\mathcal{E}_{ij})$. In addition, we add the edges $(s, p^{ij}_1), (s, q^{ij}_1), (s, \bar{r}^{ij}_1)$, $(\bar{p}^{ij}_{n+1}, t), (\bar{q}^{ij}_{n+1}, t), (\bar{r}^{ij}_{n+1}, t)$ to $\mathcal{E}_{ij}$. For each $e \in E(\mathcal{E}_{ij})$, we set $\mathsf{col}'(e) = \{2\}$. This completes the description of the edge selection gadget.

Edge Compatibility Gadget. This gadget is used to ensure that the edge selected by $\mathcal{E}_{ij}$ and $\mathcal{E}_{ji}$ corresponds to the same edge of G. For $i, j \in [k]$, $i < j$, the edge compatibility gadget $\mathcal{C}_{ij}$ is constructed as described below. Basically, $\mathcal{C}_{ij}$ comprises of a set of edges between vertices in E_{ij} and vertices in E_{ji}. Recall that E_{ij} and E_{ji} contains vertices corresponding to the same edges, namely the edges between V_i and V_j in G. Hence, we can think of E_{ji} as a set comprising of a copy of the vertices in E_{ij}. We fix a lexicographic ordering on vertices in E_{ij} which we obtain as follows. For $e_{v^i_a, v^j_x}, e_{v^i_b, v^j_y} \in E_{ij}$, $e_{v^i_a, v^j_x} < e_{v^i_b, v^j_y}$ if (i) $a < b$ or (ii) $a = b$ and $x < y$. We denote the ordering of vertices in E_{ij} by $e^{ij}_1, e^{ij}_2, \ldots, e^{ij}_m$. Refer to Figure 8 for an illustration. Note this also fixes an ordering of vertices in E_{ji} which we denote by $e^{ji}_1, e^{ji}_2, \ldots, e^{ji}_m$. Here, m is the number of edges between V_i and V_j in G. For $\ell \in [m-1]$, we add the edges $(e^{ij}_\ell, e^{ij}_{\ell+1}), (e^{ij}_\ell, e^{ji}_{\ell+1}), (e^{ji}_\ell, e^{ij}_{\ell+1}), (e^{ji}_\ell, e^{ji}_{\ell+1})$ to $\mathcal{C}_{ij}$. That is we add all the edges in the bipartition between each consecutive pair of vertices in the ordered sets E_{ij} and E_{ji}. We add edges $(s, e^{ij}_1), (s, e^{ji}_1)(e^{ij}_m, t), (e^{ji}_m, t)$ to $\mathcal{C}_{ij}$. For each edge $e \in \mathcal{C}_{ij}$, we set $\mathsf{col}'(e) = \{1\}$. We note here that in case we have created multiple edges say e, e' between vertices u, v then we delete e' and set $\mathsf{col}'(e) := \mathsf{col}'(e) \cup \mathsf{col}'(e')$.

We finally set $k' = k(k-1) + 2\binom{k}{2}$. In the following we prove certain lemmata which will be helpful in establishing the equivalence between the given instance of Multicolored Clique and the created instance of Simultaneous Cut. We denote the graph constructed above as G' with the coloring function on the edge set being col'. For $i \in [2]$, by G'_i we denote the graph with vertex set $V(G')$ and edge set $E_i = \{e \in E(G') \mid i \in \mathsf{col}'(e)\}$.

▶ **Lemma 5.1.** *For $i \in [n]$, consider the graph $\hat{G}_i = G'_1[V(\mathcal{S}_i)]$. The minimum (s,t)-separator in $\hat{G}_i$ has size $k-1$. Furthermore, $\mathcal{F} = \{V^i_\ell \mid \ell \in [n]\}$ is the set of all minimum sized (s,t)-separators in $\hat{G}_i$.*

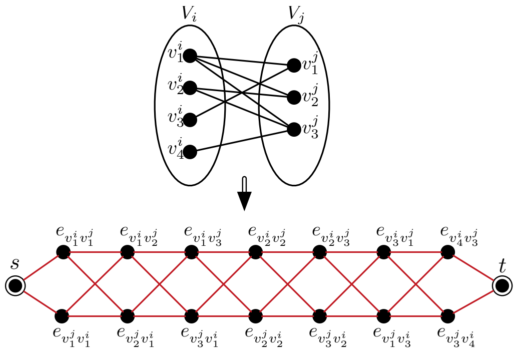

Figure 8 An illustration of edge compatibility gadget with red color denoting color 1 and $i < j$.

Proof. We start by showing that for any (s,t)-separator X' in $\hat{G}_i$, there exists $\ell \in [n]$ such that $V_\ell^i \subseteq X'$. Suppose not, then there exists an (s,t)-separator X, in $\hat{G}_i$ such that for all $\ell \in [n]$, $V_\ell^i \not\subseteq X$. This in turn implies that for all $\ell \in [n]$, there exists $v_\ell^* \in V_\ell^i$ such that $v_\ell^* \notin X$. Recall that from construction $(s, v_1^*), (v_n^*, t) \in E(\hat{G}_i)$ and for all $\ell \in [n-1]$, $(v_\ell^*, v_{\ell+1}^*) \in E(\hat{G}_i)$. This implies that there is an (s,t)-path in $\hat{G}_i - X$, namely $P = s, v_1^*, v_2^*, \ldots, v_n^*, t$, contradicting that X is an (s,t)-separator in $\hat{G}_i$. Since for any (s,t)-separator X' in $\hat{G}_i$, there exists $\ell \in [n]$ such that $V_\ell^i \subseteq X'$, this implies that the size of minimum (s,t)-separator in $\hat{G}_i$ is at least $k-1$ and $\mathcal{F} = \{V_\ell^i \mid \ell \in [n]\}$ is the set of all minimum sized (s,t)-separators in $\hat{G}_i$. ◀

▶ **Lemma 5.2.** *For $i, j \in [n]$ and $i \neq j$, consider the graph $\hat{G}_{ij} = G_1'[V(E_{ij}) \cup V(E_{ji}) \cup \{s,t\}]$. The minimum (s,t)-separator in $\hat{G}_{ij}$ has size 2. Furthermore, $\mathcal{F} = \{\{e_\ell^{ij}, e_\ell^{ji}\} \mid \ell \in [m]\}$ is the set of all minimum sized (s,t)-separators in $\hat{G}_{ij}$. Here, m is the number of edges between V_i and V_j in G.*

Proof. We start by showing that for any (s,t)-separator X' in $\hat{G}_{ij}$, there exists $\ell \in [m]$ such that $e_\ell^{ij}, e_\ell^{ji} \in X'$. Suppose not, then there exists an (s,t)-separator X, in $\hat{G}_{ij}$ such that for all $\ell \in [m]$ there exits $\hat{e}_\ell \in \{e_\ell^{ij}, e_\ell^{ji}\}$ such that $\hat{e}_\ell \notin X'$. Recall that from construction $(s, \hat{e}_1), (\hat{e}_m, t) \in E(\hat{G}_{ij})$ and for all $\ell \in [m-1]$, $(\hat{e}_\ell, \hat{e}_{\ell+1}) \in E(\hat{G}_{ij})$. This implies that there is an (s,t)-path in $\hat{G}_{ij} - X$, namely $P = s, \hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n, t$, contradicting that X is an (s,t)-separator in $\hat{G}_{ij}$. Since for any (s,t)-separator X' in $\hat{G}_{ij}$, there exists $\ell \in [m]$ such that $\{e_\ell^{ij}, e_\ell^{ji}\} \subseteq X'$, this implies that the size of minimum (s,t)-separator in $\hat{G}_{ij}$ is at least 2 and $\mathcal{F} = \{\{e_\ell^{ij}, e_\ell^{ji}\} \mid \ell \in [m]\}$ is the set of all minimum sized (s,t)-separators in $\hat{G}_{ij}$. ◀

▶ **Lemma 5.3.** *For $i, j \in [n]$ and $i \neq j$, consider the graph $\hat{G}_{ij} = G_2'[V(\mathcal{E}_{ij})]$. The minimum (s,t)-separator in $\hat{G}_{ij}$ has size 2. For $\ell \in [n]$, let $\mathcal{F}_\ell = \{\{e, v_{j\ell}^i\} \mid e \in E_\ell^{ij}\}$. Then, $\mathcal{F} = \cup_{\ell \in [n]} \mathcal{F}_\ell$ is the set of all minimum sized (s,t)-separators in $\hat{G}_{ij}$.*

Proof. We start by showing that for any (s,t)-separator X' of size at most 2 in $\hat{G}_{ij}$, there exists $\ell \in [n]$, $e \in E_\ell^{ij}$ such that $v_{j\ell}^i, e \in X'$. Suppose not. Then there exists an (s,t)-separator X, in $\hat{G}_{ij}$ of size at most 2, such that for all $\ell \in [n]$, either $v_{j\ell}^i \notin X$ or $E_\ell^{ij} \cap X = \emptyset$. Since $|X| \leq 2$, for every barrier block $\mathbf{K}_\ell^{ij}$, for $\ell \in [n+1]$, there exists $p_\ell^* \in \{p_\ell^{ij}, q_\ell^{ij}, r_\ell^{ij}\} \setminus X$ and $\bar{p}_\ell^* \in \{\bar{p}_\ell^{ij}, \bar{q}_\ell^{ij}, \bar{r}_\ell^{ij}\} \setminus X$. For each $\ell \in [n]$, we now define a path P_ℓ as follows. If $v_{j\ell}^i \notin X$ then $P_\ell = \bar{p}_\ell^*, v_{j\ell}^i, p_{\ell+1}^*$, otherwise $P_\ell = \bar{p}_\ell^*, E_\ell^{ij}, p_{\ell+1}^*$, where the path is defined with respect to the ordering of vertices of E_ℓ^{ij} used in the construction of the edge selection gadget. But then, $P = s, p_1^*, \bar{p}_1^*, P_1^{ij}, p_2^*, P_2^{ij}, \ldots, p_n^* P_n^{ij}, \bar{p}_{n+1}^*, t$ is an (s,t)-path in $\hat{G}_{ij} - X$, contradicting the assumption that X is an (s,t)-separator in $\hat{G}_{ij}$. Furthermore, it follows from the construction that for all $\ell \in [n]$, $e \in E^{ij}_{\ell}$, $\{v^i_{j\ell}, e\}$ is an (s,t)-separator in $\hat{G}_{ij}$. This concludes the proof. ◀

▶ **Lemma 5.4.** $(G, V_1, V_2, \ldots, V_k)$ *is a yes-instance of* Multicolored Clique *if and only if* $(G', s, t, k', \mathsf{col}' : E(G') \to 2^{\{1,2\}})$ *is a yes-instance of* Simultaneous OCT.

Proof. In the forward direction suppose that $(G, V_1, V_2, \ldots, V_k)$ is a yes-instance of Multicolored Clique and let $X = \{v^i_{\ell_i} \in V_i \mid i \in [k]\}$ be a set such that $G[X]$ is a clique. We note here that for $i \in [k]$, $v^i_{\ell_i}$ is the ℓ_ith vertex in V_i. Let $Y = \{e_{uv}, e_{vu} \mid u, v \in X\}$ and $X' = (\cup_{i\in[k]} V^i_{\ell_i}) \cup Y$. We will show that X' is a solution to Simultaneous Cut in $(G', s, t, k', \mathsf{col}' : E(G') \to 2^{\{1,2\}})$. Note that $|X'| = k' = k(k-1) + 2\binom{k}{2}$. Therefore, we only need to show that $G'_1 - X'$ and $G'_2 - X'$ have no (s,t)-paths, respectively.

We first show that $G'_1 - X'$ has no (s,t)-paths. Consider the set $\mathcal{Z} = \{V(\mathcal{S}_i) \setminus \{s,t\} \mid i \in [k]\} \cup \{E_{ij} \mid i,j \in [k], i \neq j\}$. Observe that for any two distinct sets $A, B \in \mathcal{Z}$, $A \cap B = \emptyset$, and for any $a \in A$ and $b \in B$, $(a,b) \notin E(G'_1)$. Hence any (s,t)-separator in G'_1 is the union of (s,t)-separators in $G'_1[A \cup \{s,t\}]$, for $A \in \mathcal{Z}$. But then from the construction of the set X', Lemma 5.1 and Lemma 5.2 it follows that X' is an (s,t)-separator in G'_1.

We will now show that $G'_2 - X'$ has no (s,t)-paths. Consider the set $\mathcal{Z}' = \{V(\mathcal{E}_{ij}) \setminus \{s,t\} \mid i,j \in [k], i \neq j\}$. For any two distinct sets $A', B' \in \mathcal{Z}'$, $A' \cap B' = \emptyset$, and for any $a' \in A'$ and $b' \in B'$, $(a',b') \notin E(G'_2)$. Hence any (s,t)-separator in G'_2 is the union of (s,t)-separators in $G'_2[A' \cup \{s,t\}]$, for $A' \in \mathcal{Z}'$. But then from the construction of the set X' and Lemma 5.3 it follows that X' is an (s,t)-separator in G'_2. This concludes the proof in the forward direction.

In the reverse direction, let $(G', s, t, k', \mathsf{col}' : E(G') \to 2^{\{1,2\}})$ be a yes-instance of Simultaneous OCT and X' be one of its solution. Since X' is a solution, therefore, $G'_1 - X'$ and $G'_2 - X'$ have no (s,t)-paths. Consider the set $\mathcal{Z} = \{V(\mathcal{S}_i) \setminus \{s,t\} \mid i \in [k]\} \cup \{E_{ij} \mid i,j \in [k], i \neq j\}$. Observe that for any two distinct sets $A, B \in \mathcal{Z}$, $A \cap B = \emptyset$, and for any $a \in A$ and $b \in B$, $(a,b) \notin E(G'_1)$. Hence, any (s,t)-separator in G'_1 is the union of (s,t)-separator in $G'_1[A \cup \{s,t\}]$, for $A \in \mathcal{Z}$. This together with Lemma 5.1, Lemma 5.2 and the definition of k' implies that for each $A \in \mathcal{Z}$, we must pick a minimum sized separator (s,t)-separator in $G'_1[A \cup \{s,t\}]$. Any minimum sized (s,t)-separator in $G'_1[V(\mathcal{S}_i)]$ must belong to $\mathcal{F}_v = \{V^i_\ell \mid \ell \in [n]\}$. We let $X = \{v^i_{\ell_i} \mid i \in [k], V^i_{\ell_i} \subseteq X'\}$. We will show that X is a solution to Multicolored Clique in $(G, V_1, V_2, \ldots, V_k)$. It is easy to see that $|X \cap V_i| = 1$. We need to show that $G[X]$ is a clique. Consider $v^i_{\ell_i}, v^j_{\ell_j} \in X$, where $i, j \in [k]$, $i < j$ and suppose $(v^i_{\ell_i}, v^j_{\ell_j}) \notin E(G)$. Lemma 5.3 (together with construction of k') implies that for some $e \in E^{ij}_{\ell_i}$, $e \in X'$ and for some $e' \in E^{ji}_{\ell_j}$, $e' \in X'$. Moreover, Lemma 5.3 implies that $\{e, e'\} \in \mathcal{F} = \{\{e^{ij}_\ell, e^{ji}_\ell\} \mid \ell \in [m]\}$. But this implies that $(v^i_{\ell_i}, v^j_{\ell_j}) \in E(G)$. This concludes the proof. ◀

Theorem 5.5 follows from combining Lemma 5.4 and the W[1]-hardness of Multicolored Clique.

▶ **Theorem 5.5.** *For all* $\alpha \geq 2$, Simultaneous Cut *is* W[1]-hard *when parameterized by* k. *Here,* α *is the number of colors in the coloring function of the edge set.*

5.2 From Simultaneous Cut to Simultaneous OCT

In this section we give a parameterized reduction from Simultaneous Cut to Simultaneous OCT. Roughly speaking, given an instance $(G, s, t, k, \mathsf{col} : E(G) \to 2^{[\alpha]})$ of Simultaneous Cut we create an instance $(G', k', \mathsf{col}' : E(G') \to 2^{[\alpha]})$ of Simultaneous OCT by *subdividing* edges in G and adding $k+1$ vertex disjoint (s,t)-paths on 2 vertices. Furthermore, we create $k+1$ duplicates (also known as *false twins*) of s and t. The objective behind these operations is the conversion of (s,t)-paths in G to odd-cycles in G'. Moreover, all the odd cycles in G' will be shown to correspond to an (s',t')-path (subdivided), where s' and t' are false twins of s and t, respectively, along with one of the newly added paths on 2 vertices. Along the way, we will also duplicate certain vertices $k+1$ times simply to ensure that a copy of these vertices always remains in the graph resulting from deleting a set of at most k vertices. We now move to the formal description of the reduction.

Let $(G, s, t, k, \mathsf{col} : E(G) \to 2^{[\alpha]})$ be an instance of Simultaneous Cut. For technical reasons we will assume that $(s,t) \notin E(G)$. Such an assumption is legitimate because otherwise, either we have a trivial no-instance of Simultaneous Cut which is the case when $\mathsf{col}((s,t)) \neq \emptyset$ or we can delete the edge (s,t) which is the case when $\mathsf{col}((s,t)) = \emptyset$. We create an instance $(G', k', \mathsf{col}' : E(G') \to 2^{[\alpha]})$ of Simultaneous OCT as follows. Initially, $V(G') = V(G)$ and $E(G') = \emptyset$. For each edge $(u,v) \in E(G)$ we add a vertex e_{uv} to $V(G')$, add the edges $(u, e_{uv}), (v, e_{uv})$ to $E(G')$, and set $\mathsf{col}'((u, e_{uv})) = \mathsf{col}((u,v))$ and $\mathsf{col}'((v, e_{uv})) = \mathsf{col}((u,v))$. For $i \in [k+1]$, we add vertices w_i, w_i' to $V(G')$ and add the edges $(s, w_i), (w_i, w_i')$, (w_i', t) to $E(G')$. In addition, we set $\mathsf{col}'((s, w_i)) = \mathsf{col}'((w_i, w_i')) = \mathsf{col}'((w_i', t)) = [\alpha]$. We create k chromatic false twins, i.e. false twins with the color sets on edges duplicated appropriately, of vertices s and t respectively in G' and let $S_f = \{s_i \mid i \in [k]\} \cup \{s\}$ and $T_f = \{t_i \mid i \in [k]\} \cup \{t\}$. Finally, we set $k' = k$.

▶ **Proposition 2.** *Let H be a graph containing a cycle C with an odd number of vertices. Then H contains an induced cycle C' with an odd number of vertices.*

In the following lemmata we establish some of the properties of the instance $(G', k', \mathsf{col}' : E(G') \to 2^{[\alpha]})$ of Simultaneous OCT that will be helpful in establishing its equivalence with the instance $(G, s, t, k, \mathsf{col} : E(G) \to 2^{[\alpha]})$ of Simultaneous Cut. We let G' to be the graph constructed as described above from G. For $i \in [\alpha]$, by G_i we denote the graph $G[E_i]$, where $E_i = \{e \in E(G) \mid i \in \mathsf{col}(e)\}$. Analogously, we define G_i', for $i \in [\alpha]$.

▶ **Lemma 5.6.** *For $i \in [\alpha]$, let C be an induced cycle in G_i' such that $|V(C)| \neq 4$. Then, $|V(C) \cap S_f| \leq 1$ and $|V(C) \cap T_f| \leq 1$.*

Proof. Consider an induced cycle C in G_i' such that $|V(C)| \neq 4$. We will only argue that $|V(C) \cap S_f| \leq 1$ and $|V(C) \cap T_f| \leq 1$ will follow from a symmetric argument. If C contains a vertex from S_f, say s', then C must contain at least 2 vertices from $\{w_i, w_i' \mid i \in [k+1]\} \cup \{e_{sv} \mid v \in N_{G_i}(s)\}$ since they are the only neighbors of s' in G'. But then C cannot contain any other vertex from S_f since vertices in S_f are chromatic false twins of s' in G' and $|V(C)| \neq 4$. This concludes the proof. ◀

▶ **Lemma 5.7.** *For $i \in [\alpha]$, let C be an induced cycle in G_i' such that $V(C) \cap \{w_i, w_i' \mid i \in [k+1]\} = \emptyset$. Then, C is a cycle with an even number of vertices.*

Proof. Consider an induced cycle C in G_i' such that $|V(C)| \cap \{w_i, w_i' \mid i \in [k+1]\} = \emptyset$. If $|V(C)| = 4$ then the claim trivially holds. Otherwise, Lemma 5.6 implies that $|V(C) \cap S_f| \leq 1$ and $|V(C) \cap T_f| \leq 1$. Since vertices in S_f and T_f are chromatic false twins, we can find a cycle C' with $|V(C)|$ vertices by replacing vertex $s' \in V(C) \cap S_f$ (if it exists) by s and vertex $t' \in V(C) \cap S_f$ (if it exists) by t. Recall that for $X = (V(G') \setminus (S_f \cup T_f \cup \{w_i, w_i' \mid i \in [k+1]\})) \cup \{s, t\}$, $G'[X]$ is a graph obtained by subdivision of edges in G. But C' is a cycle in $G_i'[X]$ and hence it follows that $|V(C')| = |V(C)|$ is an even number. ◀

▶ **Lemma 5.8.** *For $i \in [\alpha]$, let C be an induced cycle in G'_i such that $|V(C)| \neq 4$. Then for $\ell \in [k+1]$, $w_\ell \in V(C)$ if and only if $w'_\ell \in V(C)$. Furthermore, if $|V(C)| \neq 6$ then $|V(C) \cap \{w_j \mid j \in [k+1]\}| \leq 1$.*

Proof. For $\ell \in [k+1]$, let C be an induced cycle in G'_i such that $w_\ell \in V(C)$. Recall that $N_{G'_i}(w_\ell) = S_f \cup \{w'_\ell\}$. Therefore, C must contain two vertices from $S_f \cup \{w'_\ell\}$. From Lemma 5.6 it follows that $|V(C) \cap S_f| \leq 1$. This implies that $w'_\ell \in V(C)$. An analogous argument can be given for the reverse direction.

For the second part of the lemma, suppose there exists distinct $\ell, \ell' \in [k+1]$ such that $w_\ell, w_{\ell'} \in V(C)$. First part of the lemma implies that $w'_\ell, w'_{\ell'} \in V(C)$. But then C must contain a vertex $s' \in S_f$ and a vertex $t' \in T_f$ as C must contain two neighbors of w_ℓ and two neighbors of w'_ℓ. But $s' \in N_{G'_i}(w_{\ell'})$ and $t' \in N_{G'_i}(w'_{\ell'})$. This contradicts the assumption that C is an induced cycle such that $|V(C)| \neq 6$. ◀

▶ **Lemma 5.9.** *$(G, s, t, k, \mathsf{col} : E(G) \to 2^{[\alpha]})$ is a yes-instance of* Simultaneous Cut *if and only if $(G', k', \mathsf{col}' : E(G') \to 2^{[\alpha]})$ is a yes-instance of* Simultaneous OCT*.*

Proof. In the forward direction let $(G, s, t, k, \mathsf{col} : E(G) \to 2^{[\alpha]})$ be a yes-instance of Simultaneous Cut and $S \subseteq V(G) \setminus \{s, t\}$ be one of its solutions. We will show that S is a solution to the instance $(G', k', \mathsf{col}' : E(G') \to 2^{[\alpha]})$ of Simultaneous OCT. Suppose not. Then, there is an odd cycle $\hat{C}$ in $G'_i - S$, for some $i \in [\alpha]$. Since $G'_i - S$ has an odd-cycle $\hat{C}$, Proposition 2 implies that $G'_i - S$ has an induced odd-cycle C. As C is an odd-cycle, Lemma 5.7 and Lemma 5.8 imply that there exists a unique $\ell \in [k+1]$ such that $w_\ell, w'_\ell \in V(C)$. But then C must contain a vertex in S_f and a vertex in T_f. This together with Lemma 5.6 implies that there exists a unique $s' \in S_f$ and $t' \in T_f$ such that $s', t' \in V(C)$. Let P' be the path from s' to t' obtained from C by deleting w_ℓ and w'_ℓ. Since $V(P') \cap (S_f \setminus \{s'\}) = \emptyset$ and $V(P') \cap (T_f \setminus \{t'\}) = \emptyset$ it must be that all the internal vertices in P' are in $X = V(G') \setminus (S_f \cup T_f \cup \{w_j, w'_j \mid j \in [k+1]\} \cup S)$. Recall that $G'[X]$ is obtained from G by subdividing edges in G. But then we can obtain an (s,t)-path in $G_i - S$ from P' by replacing s' by s, t' by t and edges $(u, e_{uv})(e_{uv}, v)$ by (u, v) in $G - S$ contradicting that S is a solution to Simultaneous Cut.

In the reverse direction, let $(G', k', \mathsf{col}' : E(G') \to 2^{[\alpha]})$ be a yes-instance of Simultaneous OCT and $S' \subseteq V(G')$ be a solution. Let $\hat{S} = S' \setminus (S_f \cup T_f \cup \{w_i, w'_i \mid i \in [k+1]\})$. We obtain S from $\hat{S}$ by replacing each $e_{uv} \in \hat{S}$ (if any) by either of u or v. Here, in making the choice we give preference to one that is not s nor t and since $(s, t) \notin E(G)$ such a choice always exists. We will show that S is a solution to the instance $(G, s, t, k, \mathsf{col} : E(G) \to 2^{[\alpha]})$ of Simultaneous Cut. Note that $|S| \leq k$, therefore it is enough to show that for each $i \in [\alpha]$, $G_i - S$ has no (s,t)-path. Aiming for a contradiction, suppose for some $i \in [\alpha]$, $G_i - S$ has an (s,t)-path P. Since $|S'| \leq k$, there exists $j \in [k+1]$ such that $w_j, w'_j \notin S'$, $s' \in S_f \setminus S'$ and $t' \in T_f \setminus S'$. Let P'_1 be the (s',t')-path in G'_i obtained from P by replacing each edge (u, v) by (u, e_{uv}) and (e_{uv}, v), replacing s by s' and t by t'. Also, let $P'_2 = s', w_j, w'_j, t'$ be another (s',t')-path in G'_i. Recall that by construction, for each edge $e \in E(P'_1) \cup E(P'_2)$, $i \in \mathsf{col}'(e)$. Furthermore, $S' \cap (V(P'_1) \cup V(P'_2)) = \emptyset$, which follows from our construction of the paths P'_1 and P'_2. But then we have two (s',t')-paths P'_1 and P'_2 (internally vertex disjoint). Therefore, we obtain a cycle C containing s' and t' with paths P'_1 and P'_2 between them. Notice that C has an odd number of vertices since P'_2 has an even number of vertices and P'_1 has odd number of vertices. This contradicts the fact that S' is a solution to Simultaneous OCT, as needed. ◀

As a consequence of the reduction presented above, we obtain the following theorem.

▶ **Theorem 5.10.** *For all $\alpha \geq 2$,* Simultaneous OCT *is* W[1]-hard *when parameterized by k. Here, α is the number of colors in the coloring function of the edge set.*

6 Conclusion

In light of Theorem 4.8, it is natural to ask whether one can improve the running time of our algorithm for Colorful Multiway Cut. In particular, is it possible to solve the problem in $\mathcal{O}^\star(k^{\mathcal{O}(k)})$ time when the number of terminals is constant and the number of colors is at most k? Another interesting question which remains open is whether the Simultaneous FVS/OCT problem admits a (randomized) polynomial kernel. Finally, we would also like to point out another interesting consequence of Theorem 5.10, i.e. the fact that Simultaneous OCT is W[1]-hard when parameterized by k. If we replace minimal feedback vertex sets by minimal odd cycle transversals in Lemma 2.3 then Theorem 5.10 implies that such a lemma cannot be true.

Acknowledgements: The authors are thankful to Saket Saurabh for helpful discussions.

References

1 Akanksha Agrawal, Daniel Lokshtanov, Amer E. Mouawad, and Saket Saurabh. Simultaneous feedback vertex set: A parameterized perspective. In *33rd Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 7:1–7:15, 2016.
2 Leizhen Cai and Junjie Ye. Dual connectedness of edge-bicolored graphs and beyond. In *39th International Symposium on Mathematical Foundations of Computer Science*, pages 141–152, 2014.
3 Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Transactions on Algorithms*, 11(3):21:1–21:35, 2015.
4 Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016.
5 Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved Algorithms for Feedback Vertex Set Problems. *Journal of Computer and System Sciences*, 74(7):1188–1198, 2008.
6 Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved Upper Bounds for Vertex Cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010.
7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
8 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset Feedback Vertex Set Is Fixed-Parameter Tractable. *SIAM Journal of Discrete Mathematics*, 27(1):290–309, 2013.
9 Reinhard Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, 4th edition, 2010.
10 Rod G. Downey and Michael R. Fellows. *Parameterized complexity*. Springer-Verlag, 1997.
11 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, (FOCS)*, pages 470–479, 2012.
12 Robert Ganian, M. S. Ramanujan, and Stefan Szeider. Discovering archipelagos of tractability for constraint satisfaction and counting. *ACM Transactions on Algorithms*, 13(2):29:1–29:32, 2017.
13 Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72(8):1386 – 1396, 2006.

14 Yoichi Iwata, Keigo Oka, and Yuichi Yoshida. Linear-time FPT algorithms via network flow. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1749–1761, 2014.

15 Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811, 2014.

16 Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Transactions on Algorithms*, 12(2):21:1–21:41, 2016.

17 Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic Feedback Vertex Set. *Information Processing Letters*, 114(10):556–560, 2014.

18 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster Parameterized Algorithms Using Linear Programming. *ACM Transactions on Algorithms*, 11(2):15:1–15:31, 2014.

19 Daniel Lokshtanov and M. S. Ramanujan. Parameterized Tractability of Multiway Cut with Parity Constraints. In *39th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 750–761, 2012.

20 Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. A linear time parameterized algorithm for directed feedback vertex set. *ArXiv e-prints*, 2016. `arXiv:1609.04347`.

21 Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms*, 9(4):30:1–30:35, 2013.

22 M. S. Ramanujan and Saket Saurabh. Linear time parameterized algorithms via skew-symmetric multicuts. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1739–1748, 2014.

23 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

24 Junjie Ye. A note on finding dual feedback vertex set. *ArXiv e-prints*, 2015. `arXiv:1510.00773`.