NODE MULTIWAY CUT AND SUBSET FEEDBACK VERTEX SET ON GRAPHS OF BOUNDED MIM-WIDTH

BERGOUGNOUX BENJAMIN, CHARIS PAPADOPOULOS, AND JAN ARNE TELLE

Abstract. The two weighted graph problems Node Multiway Cut (NMC) and Subset Feedback Vertex Set (SFVS) both ask for a vertex set of minimum total weight, that for NMC disconnects a given set of terminals, and for SFVS intersects all cycles containing a vertex of a given set. We design a meta-algorithm that will allow to solve both problems in time $2^{O(rw^3)} \cdot n^4$, $2^{O(q^2 \log(q))} \cdot n^4$, and $n^{O(k^2)}$ where rw is the rank-width, q the $\mathbb{Q}$-rank-width, and k the mim-width of a given decomposition. This answers in the affirmative an open question raised by Jaffke et al. (Algorithmica, 2019) concerning an XP algorithm for SFVS parameterized by mim-width.

By a unified algorithm, this solves both problems in polynomial-time on the following graph classes: Interval, Permutation, and Bi-Interval graphs, Circular Arc and Circular Permutation graphs, Convex graphs, k-Polygon, Dilworth-k and Co-k-Degenerate graphs for fixed k; and also on Leaf Power graphs if a leaf root is given as input, on H-Graphs for fixed H if an H-representation is given as input, and on arbitrary powers of graphs in all of the above classes. Prior to our results, only SFVS was known to be tractable restricted only on Interval and Permutation graphs, whereas all other results are new.

1. Introduction

Given a vertex-weighted graph G and a set S of its vertices, the Subset Feedback Vertex Set (SFVS) problem asks for a vertex set of minimum weight that intersects all cycles containing a vertex of S. SFVS was introduced by Even et al. [17] who proposed an 8-approximation algorithm. Cygan et al. [15] and Kawarabayashi and Kobayashi [28] independently showed that SFVS is fixed-parameter tractable (FPT) parameterized by the solution size, while Hols and Kratsch [24] provide a randomized polynomial kernel for the problem. As a generalization of the classical NP-complete Feedback Vertex Set (FVS) problem, for which $S = V(G)$, there has been a considerable amount of work to obtain faster algorithms for SFVS, both for general graphs where the current best is an $O^*(1.864^n)$ algorithm due to Fomin et al. [19], and restricted to special graph classes [11, 18, 22, 35, 36]. Naturally, FVS and SFVS differ in complexity, as exemplified by split graphs where FVS is polynomial-time solvable [12] whereas SFVS remains NP-hard [19]. Moreover note that the vertex-weighted variation of SFVS behaves differently that the unweighted one, as exposed on graphs with bounded independent set sizes: weighted SFVS is NP-complete on graphs with independent set size at most four, whereas unweighted SFVS is in XP parameterized by the independent set size [35].

Closely related to SFVS is the NP-hard Node Multiway Cut (NMC) problem in which we are given a vertex-weighted graph G and a set T of (terminal) vertices, and asked to find a vertex set of minimum weight that disconnects all the terminals [9, 21]. NMC was introduced by Garg et al. in 1994 [21] as a variant in which the terminals were not allowed to be deleted [21], and both variations are well-studied problems in terms of approximation, as well as parameterized algorithms [9, 11, 10, 15, 16, 19]. It is not difficult to see that SFVS for $S = \{v\}$ coincides with NMC in which $T = N(v)$. In fact, NMC reduces to SFVS by adding a single vertex v with a large weight that is adjacent to all terminals such that $S = \{v\}$ [19] (if non-deletable terminals then add a large weight to them all). Thus, in order to solve NMC on a given graph one may apply a known algorithm for SFVS on a vertex-extended graph. Observe, however, that through such an approach one needs to clarify that the vertex-extended graph still obeys the necessary properties of the known algorithm for SFVS. Therefore, despite the few positive results for SFVS on graph families [35, 36], the complexity of NMC restricted to special graph classes remained unresolved.

In this paper, we investigate the complexity of SFVS and NMC when parameterized by structural graph width parameters. Well-known graph width parameters include tree-width [6], clique-width [14], rank-width [31], and maximum induced matching width (*a.k.a.* mim-width) [37]. These are of varying strength, with tree-width of modeling power strictly weaker than clique-width, as it is bounded on a strict subset of the graph classes having bounded clique-width, with rank-width and clique-width of the same modeling power, and with mim-width much stronger than clique-width. Belmonte and Vatshelle [1] showed that several graph classes, like interval graphs and permutation graphs, have bounded mim-width and a decomposition witnessing this can be found in polynomial time, whereas it is known that the clique-width of such graphs can be proportional to the square root of the number of vertices [23]. In this way, an XP algorithm parameterized by mim-width has the feature of unifying several algorithms on well-known graph classes.

We give a meta-algorithm that for an input graph G will give parameterized algorithms for several width measures at once, by assuming that we are given a branch decomposition over the vertex set of G. This is a natural hierarchical clustering of G, represented as a subcubic tree T with the vertices of G at its leaves. Any edge of the tree defines a cut of G given by the leaves of the two subtrees that result from removing the edge from T. Judiciously choosing a cut-function to measure the complexity of such cuts, or rather of the bipartite subgraphs of G given by the edges crossing the cuts, this framework then defines a graph width parameter by a minmax relation, minimum over all trees and maximum over all its cuts. Several graph width parameters have been defined this way, like carving-width, maximum matching-width, boolean-width etc. We will in this paper focus on: (i) rank-width [31] whose cut function is the GF[2]-rank of the adjacency matrix, (ii) $\mathbb{Q}$-rank-width [33] a variant of rank-width with interesting algorithmic properties which instead uses the rank over the rational field, and (iii) mim-width [37] whose cut function is the size of a maximum induced matching of the graph crossing the cut. Note that in contrast to e.g. clique-width, for rank-width and $\mathbb{Q}$-rank-width there is a $2^{3k} \cdot n^4$ algorithm that, given a graph and $k \in \mathbb{N}$, either outputs a decomposition of width at most $3k + 1$ or confirms that the width of the input graph is more than k [33, 34].

Let us mention what is known regarding the complexity of NMC and SFVS parameterized by these width measures. Standard algorithmic techniques give a $k^{O(k)} \cdot n$ time algorithm parameterized by the tree-width of the input graph, so for this reason we do not focus on tree-width. The runtime of our meta-algorithm as a function of a given clique-width expression will be $2^{O(k^2)} \cdot n^{O(1)}$ but we think a faster runtime is achievable through known techniques [4], so we do not focus on clique-width. Since these problems can be expressed in MSO_1-logic it follows that they are FPT parameterized by clique-width, rank-width or $\mathbb{Q}$-rank-width [13, 32], however the runtime will contain a tower of 2's with more than 4 levels. Moreover, FVS and also SFVS are W[1]-hard when parameterized by the mim-width of a given decomposition [27].

Attacking SFVS seems to be a hard task that behaves differently than the already-known approaches for FVS. Even for very small values of mim-width that captures several graph classes, the tractability of SFVS, prior to our result, was left open besides interval and permutation graphs [36]. Although FVS was known to be tractable on such graphs for more than a decade [30], the complexity status of SFVS still remained unknown.

Our results. We resolve in the affirmative the question raised by Jaffke et al. [27], also mentioned in [36] and [35], asking whether there is an XP-time algorithm for SFVS parameterized by the mim-width of a given decomposition. For rank-width and $\mathbb{Q}$-rank-width we provide the first explicit FPT-algorithms with low exponential dependency that avoid the MSO_1 formulation. Our main results are summarized in the following theorem:

Theorem 1. *Let G be a graph on n vertices. We can solve* Subset Feedback Vertex Set *and* Node Multiway Cut *in time $2^{O(rw^3)} \cdot n^4$ and $2^{O(q^2 \log(q))} \cdot n^4$, where rw and q are the rank-width and the $\mathbb{Q}$-rank-width of G, respectively. Moreover, if a branch decomposition of* *mim-width k for G is given as input, we can solve* Subset Feedback Vertex Set *and* Node Multiway Cut *in time $n^{O(k^2)}$.*

Note it is not known whether the mim-width of a graph can be approximated within a constant factor in time $n^{f(k)}$ for some function f. However, by the previously mentioned results of Belmonte and Vatshelle [1] on computing decompositions of bounded mim-width, combined with a result of [25] showing that for any positive integer r a decomposition of mim-width k of a graph G is also a decomposition of mim-width at most $2k$ of its power G^r, we get the following corollary.

Corollary 2. *We can solve* Subset Feedback Vertex Set *and* Node Multiway Cut *in polynomial time on* Interval*,* Permutation*, and* Bi-Interval *graphs,* Circular Arc *and* Circular Permutation *graphs,* Convex *graphs, k-*Polygon*,* Dilworth-*k and* Co-*k-*Degenerate *graphs for fixed k, and on arbitrary powers of graphs in any of these classes.*

Previously, such polynomial-time tractability was known only for SFVS and only on Interval and Permutation graphs [36]. It is worth noticing that Theorem 1 implies also that we can solve Subset Feedback Vertex Set and Node Multiway Cut in polynomial time on Leaf Power (from which we can compute a decomposition of mim-width 1) if an intersection model is given as input [1, 25] and on H-Graphs for a fixed H if an H-representation is given as input (from which we can compute a decomposition of mim-width $2|E(H)|$) [20].

Our approach. We give some intuition to our meta-algorithm, that will focus on Subset Feedback Vertex Set. This since NMC can be solved by adding a vertex v of large weight adjacent to all terminals and solving SFVS with $S = \{v\}$, all within the same runtime as extending the given branch decomposition to this new graph increases the width at most by one for all considered width measures.

Towards achieving our goal, we use d-neighbor equivalence, for $d = 1$ and $d = 2$, a notion introduced by Bui-Xuan et al. [8]. Two subsets X and Y of $A \subseteq V(G)$ are *d-neighbor equivalent w.r.t. A*, if $\mathsf{min}(d, |X \cap N(u)|) = \mathsf{min}(d, |Y \cap N(u)|)$ for all $u \in V(G) \setminus A$. For a cut $(A, \overline{A})$ this equivalence relation on subsets of vertices was used by Bui-Xuan et al. [8] to design a meta-algorithm, also giving XP algorithms by mim-width, for so-called (σ, ρ) generalized domination problems. Recently, Bergougnoux and Kanté [5] extended the uses of this notion to acyclic and connected variants of (σ, ρ) generalized domination and similar problems like FVS. An earlier XP algorithm for FVS parameterized by mim-width had been given by Jaffke et al. [27] but instead of the d-neighbor equivalences this algorithm was based on reduced forests and minimal vertex covers.

Our meta-algorithm does a bottom-up traversal of a given branch decomposition of the input graph G, computing a vertex subset X of maximum weight that induces an *S-forest* (i.e., a graph where no cycle contains a vertex of S) and outputs $V(G) \setminus X$ which is necessarily a solution of SFVS. As usual, our dynamic programming algorithm relies on a notion of representativity between sets of partial solutions. For each cut $(A, \overline{A})$ induced by the decomposition, our algorithm computes a set of partial solutions $\mathcal{A} \subseteq 2^A$ of small size that represents 2^A. We say that a set of partial solutions $\mathcal{A} \subseteq 2^A$ represents a set of partial solutions $\mathcal{B} \subseteq 2^A$, if, for each $Y \subseteq \overline{A}$, we have $\mathsf{best}(\mathcal{A}, Y) = \mathsf{best}(\mathcal{B}, Y)$ where $\mathsf{best}(\mathcal{C}, Y)$ is the maximum weight of a set $X \in \mathcal{C}$ such that $X \cup Y$ induces an S-forest. Our main tool is a subroutine that, given a set of partial solutions $\mathcal{B} \subseteq 2^A$, outputs a subset $\mathcal{A} \subseteq \mathcal{B}$ of small size that represents $\mathcal{B}$.

To design this subroutine, we cannot use directly the approaches solving FVS of any earlier approaches, like [5] or [27]. This is due to the fact that S-forests behave quite differently than forests; for example, given an S-forest F, the graph induced by the edges between $A \cap V(F)$ and $\overline{A} \cap V(F)$ could be a biclique. Instead, we introduce a notion of vertex contractions and prove that, for every $X \subseteq A$ and $Y \subseteq \overline{A}$, the graph induced by $X \cup Y$ is an S-forest if and only if there exists a partition of $X \setminus S$ and of $Y \setminus S$, satisfying certain properties, such that contracting the blocks of these partitions into single vertices transforms the S-forest into a forest.

This equivalence between S-forests in the original graph and forests in the contracted graphs allows us to adapt some ideas from [5] and [27]. Most of all, we use the property that, if the mim-width of the given decomposition is mim, then the contracted graph obtained from the bipartite graph induced by X and Y admits a vertex cover VC of size at most $4mim$. Note however, that in our case the elements of VC are contracted subsets of vertices. Such a vertex cover allows to control the cycles which are crossing the cut.

We associate each possible vertex cover VC with an index i which contains basically a representative for the 2-neighbor equivalence for each element in VC. Moreover, for each index i, we introduce the notions of partial solutions and complements solutions associated with i which correspond, respectively, to subsets of $X \subseteq A$ and subsets $Y \subseteq \overline{A}$ such that, for some contractions of X and Y, the contracted graph obtained from the bipartite graph induced by X and Y admits a vertex cover VC associated with i. We define an equivalence relation $\sim_i$ between the partial solutions associated with i such that $X \sim_i W$, if X and W connect in the same way the representatives of the vertex sets which belongs to the vertex covers described by i. Given a set of partial solutions $\mathcal{B} \subseteq 2^A$, our subroutine outputs a set $\mathcal{A}$ that contains, for each index i and each equivalence class $\mathcal{C}$ of $\sim_i$ over $\mathcal{B}$, a partial solution in $\mathcal{C}$ of maximum weight. In order to prove that $\mathcal{A}$ represents $\mathcal{B}$, we show that:

- for every maximum S-forest F, there exists an index i such that $V(F) \cap A$ is a partial solution associated with i and $V(F) \cap \overline{A}$ is a complement solutions associated with i.
- if $X \sim_i W$, then, for every complement solution Y associated with i, the graph induced by $X \cup Y$ is an S-forest if and only if $W \cup Y$ induces an S-forest.

The number of indices i is upper bounded by $2^{O(q^2 \log(q))}$, $2^{O(rw^3)}$ and $n^{O(mim^2)}$. This follows from the known upper-bounds on the number of 2-neighbor equivalence classes and the fact that the vertex covers we consider have size at most $4mim$. Since there are at most $(4mim)^{4mim}$ ways of connecting $4mim$ vertices and $rw, q \geq mim$, we deduce that the size of $\mathcal{A}$ is upper bounded by $2^{O(q^2 \log(q))}$, $2^{O(rw^3)}$ and $n^{O(mim^2)}$.

To the best of our knowledge, this is the first time vertex contractions are used in a dynamic programming algorithm parameterized by graph width measures. Note that in contrast to the meta-algorithms in [5, 8], the number of representatives (for the d-neighbor equivalence) contained in the indices of our meta-algorithm are not upper bounded by a constant but by $4mim$. This explains the differences between the runtimes in Theorem 1 and those obtained in [5, 8], e.g. $n^{O(mim^2)}$ versus $n^{O(mim)}$. In case $S = V(G)$, thus solving FVS, our meta-algorithm will have runtime $n^{O(mim)}$, as the algorithms for FVS of [5, 27]. We do not expect that SFVS can be solved as fast as FVS when parameterized by graph width measures. For example, given a graph of tree-width k, FVS can be solved in $2^{O(k)} \cdot n$ [7] but it seems that SFVS cannot be solved in $k^{o(k)} \cdot n^{O(1)}$ unless ETH fails [2].

2. Preliminaries

The size of a set V is denoted by $|V|$ and its power set is denoted by 2^V. We write $A \setminus B$ for the set difference of A from B. We let $\mathsf{min}(\varnothing) = +\infty$ and $\mathsf{max}(\varnothing) = -\infty$.

Graphs. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$. An edge between two vertices x and y is denoted by xy (or yx). Given $\mathcal{S} \subseteq 2^{V(G)}$, we denote by $V(\mathcal{S})$ the set $\bigcup_{S \in \mathcal{S}} S$. For a vertex set $U \subseteq V(G)$, we denote by $\overline{U}$ the set $V(G) \setminus U$. The set of vertices that are adjacent to x is denoted by $N_G(x)$, and for $U \subseteq V(G)$, we let $N_G(U) = (\cup_{v \in U} N_G(v)) \setminus U$. We will omit the subscript G whenever there is no ambiguity.

The subgraph of G induced by a subset X of its vertex set is denoted by $G[X]$. For two disjoint subsets X and Y of $V(G)$, we denote by $G[X, Y]$ the bipartite graph with vertex set $X \cup Y$ and edge set $\{xy \in E(G) \mid x \in X \text{ and } y \in Y\}$. We denote by $M_{X,Y}$ the adjacency matrix between X and Y, i.e., the (X, Y)-matrix such that $M_{X,Y}[x, y] = 1$ if $y \in N(x)$ and 0 otherwise. A *vertex cover* of a graph G is a set of vertices $\mathsf{VC} \subseteq V(G)$ such that, for every edge $uv \in E(G)$, we have $u \in \mathsf{VC}$ or $v \in \mathsf{VC}$. A *matching* is a set of edges having no common endpoint and an *induced matching* is a matching M of edges such that $G[V(M)]$ has no other edges besides M. The *size of an induced matching* M refers to the number of edges in M.

For a graph G, we denote by $\mathsf{CC}_G(X)$ the partition $\{C \subseteq V(G) \mid G[C]$ is a connected component of $G[X]\}$. For two graphs G_1 and G_2, we denote by $G_1 - G_2$ the graph $(V(G_1), E(G_1) \setminus E(G_2))$.

Given a graph G and $S \subseteq V(G)$, we say that a cycle of G is an *S-cycle* if it contains a vertex in S. Moreover, we say that a subgraph F of G is an *S-forest* if F does not contain an S-cycle. Typically, the Subset Feedback Vertex Set problem asks for a vertex set of minimum (weight) size such that its removal results in an S-forest. Here we focus on the equivalent formulation of computing a maximum weighted S-forest, formally defined as follows:

Subset Feedback Vertex Set (SFVS)
Input: A graph G, $S \subseteq V(G)$ and a weight function $\mathsf{w} : V(G) \to \mathbb{Q}$.
Output: The maximum among the weights of the S-forests of G.

Rooted Layout. A *rooted binary tree* is a binary tree with a distinguished vertex called the *root*. Since we manipulate at the same time graphs and trees representing them, the vertices of trees will be called *nodes*.

A *rooted layout* of G is a pair $\mathcal{L} = (T, \delta)$ of a rooted binary tree T and a bijective function δ between $V(G)$ and the leaves of T. For each node x of T, let L_x be the set of all the leaves l of T such that the path from the root of T to l contains x. We denote by $V_x^{\mathcal{L}}$ the set of vertices that are in bijection with L_x, i.e., $V_x^{\mathcal{L}} := \{v \in V(G) \mid \delta(v) \in L_x\}$. When $\mathcal{L}$ is clear from the context, we may remove $\mathcal{L}$ from the superscript.

All the width measures dealt with in this paper are special cases of the following one, where the difference in each case is the used set function. Given a set function $\mathsf{f} : 2^{V(G)} \to \mathbb{N}$ and a rooted layout $\mathcal{L} = (T, \delta)$, the f-width of a node x of T is $\mathsf{f}(V_x^{\mathcal{L}})$ and the f-width of (T, δ), denoted by $\mathsf{f}(T, \delta)$ (or $\mathsf{f}(\mathcal{L})$), is $\max\{\mathsf{f}(V_x^{\mathcal{L}}) \mid x \in V(T)\}$. Finally, the f-width of G is the minimum f-width over all rooted layouts of G.

($\mathbb{Q}$)-Rank-width. The rank-width and $\mathbb{Q}$-rank-width are, respectively, the rw-width and $\mathsf{rw}_{\mathbb{Q}}$-width where $\mathsf{rw}(A)$ (resp. $\mathsf{rw}_{\mathbb{Q}}(A)$) is the rank over $GF(2)$ (resp. $\mathbb{Q}$) of the matrix $M_{A,\overline{A}}$ for all $A \subseteq V(G)$.

Mim-width. The mim-width of a graph G is the mim-width of G where $\mathsf{mim}(A)$ is the size of a maximum induced matching of the graph $G[A, \overline{A}]$ for all $A \subseteq V(G)$.

Observe that all three parameters rw-, $\mathsf{rw}_{\mathbb{Q}}$-, and mim-width are symmetric, i.e., for the associated set function f and for any $A \subseteq V(G)$, we have $f(A) = f(\overline{A})$. The following lemma provides upper bounds between mim-width and the other two parameters.

Lemma 3 ([37]). *Let G be a graph. For every $A \subseteq V(G)$, we have $\mathsf{mim}(A) \leqslant \mathsf{rw}(A)$ and $\mathsf{mim}(A) \leqslant \mathsf{rw}_{\mathbb{Q}}(A)$.*

Proof. Let $A \subseteq V(G)$. Let S be the vertex set of a maximum induced matching of the graph $G[A, \overline{A}]$. By definition, we have $\mathsf{mim}(A) = |S \cap A| = |S \cap \overline{A}|$. Observe that the restriction of the matrix $M_{A,\overline{A}}$ to rows in $S \cap A$ and columns in $S \cap \overline{A}$ is the identity matrix. Hence, $\mathsf{mim}(A)$ is upper bounded both by $\mathsf{rw}(A)$ and $\mathsf{rw}_{\mathbb{Q}}(A)$. □

d-neighbor-equivalence. The following concepts were introduced in [8]. Let G be a graph. Let $A \subseteq V(G)$ and $d \in \mathbb{N}^+$. Two subsets X and Y of A are *d-neighbor equivalent w.r.t. A*, denoted by $X \equiv_A^d Y$, if $\mathsf{min}(d, |X \cap N(u)|) = \mathsf{min}(d, |Y \cap N(u)|)$ for all $u \in \overline{A}$. It is not hard to check that $\equiv_A^d$ is an equivalence relation. See Figure 1 for an example of 2-neighbor equivalent sets.

From the definition of the 2-neighbor equivalence relation we have the following.

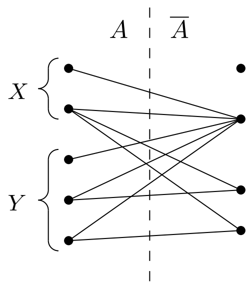


FIGURE 1. We have $X \equiv_A^2 Y$, but it is not the case that $X \equiv_A^3 Y$.

Fact 4. *For every $A \subseteq V(G)$ and $W, Z \subseteq A$, if $W \equiv_A^2 Z$, then, for all $v \in \overline{A}$, we have $|N(v) \cap W| \leqslant 1$ if and only if $|N(v) \cap Z| \leqslant 1$.*

For all $d \in \mathbb{N}^+$, we let $\mathsf{nec}_d : 2^{V(G)} \to \mathbb{N}$ where for all $A \subseteq V(G)$, $\mathsf{nec}_d(A)$ is the number of equivalence classes of $\equiv_A^d$. Notice that while nec_1 is a symmetric function [29, Theorem 1.2.3], nec_d is not necessarily symmetric for $d \geq 2$.

The following lemma shows how $\mathsf{nec}_d(A)$ is upper bounded by the other parameters.

Lemma 5 ([1, 33, 37])**.** *Let G be a graph. For every $A \subseteq V(G)$ and $d \in \mathbb{N}^+$, we have the following upper bounds on $\mathsf{nec}_d(A)$:*

(a) $2^{d \cdot \mathsf{rw}(A)^2}$, *(b)* $(d \cdot \mathsf{rw}_{\mathbb{Q}}(A) + 1)^{\mathsf{rw}_{\mathbb{Q}}(A)}$, *(c)* $|A|^{d \cdot \mathsf{mim}(A)}$.

In order to manipulate the equivalence classes of $\equiv_A^d$, one needs to compute a representative for each equivalence class in polynomial time. This is achieved with the following notion of a representative. Let G be a graph with an arbitrary ordering of $V(G)$ and let $A \subseteq V(G)$. For each $X \subseteq A$, let us denote by $\mathsf{rep}_A^d(X)$ the lexicographically smallest set $R \subseteq A$ such that $|R|$ is minimized and $R \equiv_A^d X$. Moreover, we denote by $\mathcal{R}_A^d$ the set $\{\mathsf{rep}_A^d(X) \mid X \subseteq A\}$. It is worth noticing that the empty set always belongs to $\mathcal{R}_A^d$, for all $A \subseteq V(G)$ and $d \in \mathbb{N}^+$. Moreover, we have $\mathcal{R}_{V(G)}^d = \mathcal{R}_\varnothing^d = \{\varnothing\}$ for all $d \in \mathbb{N}^+$. In order to compute these notions, we use the following lemma.

Lemma 6 ([8])**.** *Let G be an n-vertex graph. For every $A \subseteq V(G)$ and $d \in \mathbb{N}^+$, one can compute in time $O(\mathsf{nec}_d(A) \cdot n^2 \cdot \log(\mathsf{nec}_d(A)))$, the sets $\mathcal{R}_A^d$ and a data structure that, given a set $X \subseteq A$, computes $\mathsf{rep}_A^d(X)$ in time $O(|A| \cdot n \cdot \log(\mathsf{nec}_d(A)))$.*

Vertex Contractions. In order to deal with SFVS, we will use the ideas of the algorithms for FEEDBACK VERTEX SET from [5, 26]. To this end, we will contract subsets of $\overline{S}$ in order to transform S-forests into forests.

In order to compare two partial solutions associated with $A \subseteq V(G)$, we define an auxiliary graph in which we replace contracted vertices by their representative sets in $\mathcal{R}_A^2$. Since the sets in $\mathcal{R}_A^2$ are not necessarily pairwise disjoint, we will use the following notions of graph "induced" by collections of subsets of vertices. We will also use these notions to define the contractions we make on partial solutions.

Let G be a graph. Given $\mathcal{A} \subseteq 2^{V(G)}$, we define $G[\mathcal{A}]$ as the graph with vertex set $\mathcal{A}$ where $A, B \in \mathcal{A}$ are adjacent if and only if $N(A) \cap B \neq \varnothing$. Observe that $G[\mathcal{A}]$ is obtained from an induced subgraph of G by *vertex contractions* and, for this reason, we refer to $G[\mathcal{A}]$ as a *contracted graph*. In what follows, the neighborhood notation refers to the graph G and, thus, we omit the corresponding subscript G. Given $\mathcal{A}, \mathcal{B} \subseteq 2^{V(G)}$, we denote by $G[\mathcal{A}, \mathcal{B}]$ the bipartite graph with vertex set $\mathcal{A} \cup \mathcal{B}$ and where $A, B \in \mathcal{A} \cup \mathcal{B}$ are adjacent if and only if $A \in \mathcal{A}$, $B \in \mathcal{B}$, and $N(A) \cap B \neq \varnothing$. Moreover, we denote by $G[\mathcal{A} \mid \mathcal{B}]$ the graph with vertex set $\mathcal{A} \cup \mathcal{B}$ and with edge set $E(G[\mathcal{A}]) \cup E(G[\mathcal{A}, \mathcal{B}])$. Observe that both graphs $G[\mathcal{A}, \mathcal{B}]$ and $G[\mathcal{A} \mid \mathcal{B}]$ are subgraphs of the contracted graph $G[\mathcal{A} \cup \mathcal{B}]$. To avoid confusion with the original graph, we refer to the vertices of the contracted graphs as *blocks*. It is worth noticing that in the contracted graphs used in this paper, whenever two blocks are adjacent, they are disjoint.

The following fact states that we can contract from a partition without increasing the size of a maximum induced matching of a graph. It follows directly from the definition of contractions.

Fact 7. *Let H be a graph. For any partition $\mathcal{P}$ of a subset of $V(H)$, the size of a maximum induced matching of $H[\mathcal{P}]$ is at most the size of a maximum induced matching of H.*

Let (G, S) be an instance of SFVS. The vertex contractions that we use on a partial solution X are defined from a given partition of $X \setminus S$. A partition of the vertices of $X \setminus S$ is called an $\overline{S}$*-contraction* of X. We will use the following notations to handle these contractions.

Given $X \subseteq V(G)$, we denote by $\binom{X}{1}$ the partition of X which contains only singletons, i.e., $\binom{X}{1} = \bigcup_{v \in X}\{v\}$. Moreover, for an $\overline{S}$-contraction $\mathcal{P}$ of X, we denote by $X_{\downarrow \mathcal{P}}$ the partition of X where $X_{\downarrow \mathcal{P}} = \mathcal{P} \cup \binom{X \cap S}{1}$. Given a subgraph G' of G such that $V(G') = X$, we denote by $G'_{\downarrow \mathcal{P}}$ the graph $G'[X_{\downarrow \mathcal{P}}]$. It is worth noticing that in our contracted graphs, every block of S-vertices are singletons and we denote them by $\{v\}$.

Given a set $X \subseteq V(G)$, we will intensively use the graph $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$ which corresponds to the graph obtained from $G[X]$ by contracting the connected components of $G[X \setminus S]$. Observe that, for every subset $X \subseteq V(G)$, if $G[X]$ is an S-forest, then $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$ is a forest. The converse is not true as we may delete S-cycles with contractions: take a triangle with one vertex v in S and contract the neighbors of v. However, we can prove the following equivalence (see Figure 2 for an example).

Fact 8. *Let G be a graph and $S \subseteq V(G)$. For every $X \subseteq V(G)$ such that $|N(v) \cap C| \leqslant 1$ for each $v \in X \cap S$ and each $C \in \mathsf{cc}_G(X \setminus S)$, we have $G[X]$ is an S-forest if and only if $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$ is a forest.*

Proof. Suppose first that $G[X]$ is an S-forest. Assume for contradiction that there is a cycle C^* in $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$. If there is no block in C^* then all vertices of C^* belong to S and $G[C^*]$ is an S-cycle in $G[X]$. For every block $U \in C_S$, let v_1, v_2 be the two distinct neighbors of U in C^* and let C be the connected component of $G[X \setminus S]$ that corresponds to U. Observe that both v_1, v_2 belong to S. By construction, there are vertices (not necessarily distinct) u_1, u_2 such that $u_1 \in N(v_1) \cap C$ and $u_2 \in N(v_2) \cap C$. As C is a connected component, there is a path P (not necessarily non-empty) between u_1 and u_2 passing through vertices of C. Thus, replacing every block $U \in C_S$ by its corresponding path P results in an S-cycle in $G[X]$, leading to a contradiction.

For the other direction, assume for contradiction that there is an S-cycle C_S in $G[X]$. We will construct a cycle in $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$. Let $(u_1, \ldots, u_t)$ be the subpath of C_S that contains vertices of the same connected component C in $G[X \setminus S]$. Since C_S is an S-cycle, there are vertices $v, v' \in C_S \cap S$ that are neighbors of u_1 and u_t, respectively. By the fact that $|N(v) \cap C| \leqslant 1$ for every $v \in X \cap S$, we deduce that v and v' are distinct vertices in $G[X]$. Moreover, by construction we know that v and v' are distinct vertices in $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$, as well. Thus, we replace every subpath of $C_S \cap (X \setminus S)$ by its corresponding block of $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$ and construct a cycle in $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$, leading to a contradiction. Therefore, if $G[X]_{\downarrow \mathsf{cc}_G(X \setminus S)}$ is a forest then $G[X]$ is an S-forest. □

3. A meta-algorithm for Subset Feedback Vertex Set

In the following, we present a meta-algorithm that, given a rooted layout (T, δ) of G, solves SFVS. We will show that such a meta-algorithm will imply that SFVS can be solved in time $\mathsf{rw}_{\mathbb{Q}}(G)^{O(\mathsf{rw}_{\mathbb{Q}}(G)^2)} \cdot n^4$, $2^{O(\mathsf{rw}(G)^3)} \cdot n^4$ and $n^{O(\mathsf{mim}(T,\delta)^2)}$. The main idea of this algorithm is to use $\overline{S}$-contractions in order to employ similar properties of the meta-algorithm for Maximum Induced Tree of [5] and the $n^{O(\mathsf{mim}(T,\delta))}$ time algorithm for Feedback Vertex Set of [26]. In particular, we use the following lemma which is proved implicitly in [3, Lemma 5.5].

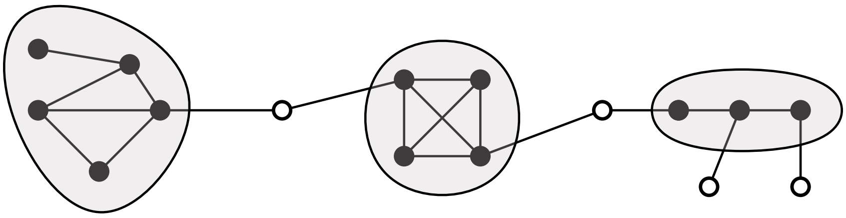


FIGURE 2. Example of an $\overline{S}$-contraction. The vertices of S are white filled.

Lemma 9. *Let X and Y be two disjoint subsets of $V(G)$. If $G[X \cup Y]$ is a forest, then the number of vertices of X that have at least two neighbors in Y is bounded by $2w$ where w is the size of a maximum induced matching in the bipartite graph $G[X, Y]$.*

Proof. Let X^{2+} be the set of vertices in X having at least 2 neighbors in Y. In the following, we prove that $F = G[X^{2+}, Y]$ admits a *good bipartition*, that is a bipartition $\{X_1, X_2\}$ of X^{2+} such that, for each $i \in \{1, 2\}$ and, for each $v \in X_i$, there exists $y_v \in Y \cap V(F)$ such that $N_F(y_v) \cap X_i = \{v\}$. Observe that this is enough to prove the lemma since if F admits a good bipartition $\{X_1, X_2\}$, then $|X_1| \leqslant w$ and $|X_2| \leqslant w$. Indeed, if F admits a good bipartition $\{X_1, X_2\}$, then, for each $i \in \{1, 2\}$, the set of edges $M_i = \{vy_v \mid v \in X_i\}$ is an induced matching of $G[X, Y]$. In order to prove that F admits a good bipartition it is sufficient to prove that each connected component of F admits a good bipartition.

Let C be a connected component of F and $u \in C \cap X^{2+}$. As $G[X \cup Y]$ is a forest, we deduce that $F[C]$ is a tree. Observe that the distance in $F[C]$ between each vertex $v \in C \cap X^{2+}$ and u is even because $F[C]$ is bipartite w.r.t. $(C \cap X^{2+}, C \setminus X^{2+})$. Let C_1 (resp. C_2) be the set of all vertices $v \in C \cap X^{2+}$ such that there exists an odd (resp. even) integer $\ell \in \mathbb{N}$ so that the distance between v and u in F is 2ℓ. We claim that $\{C_1, C_2\}$ is a good bipartition of $F[C]$.

Let $i \in \{1, 2\}$, $v \in C_i$ and $\ell \in \mathbb{N}$ such that the distance between v and u in $F[C]$ is 2ℓ. Let P be the set of vertices in $C \setminus \{v\}$ that share a common neighbor with v in $F[C]$. We want to prove that there exists $y \in Y$ such that $N_{F[C]}(y) \cap C_i = \{v\}$. For doing so, it is sufficient to prove that $N_{F[C]}(v) \setminus N_{F[C]}(C_i \setminus \{v\}) = N_{F[C]}(v) \setminus N_{F[C]}(P \cap C_i) \neq \varnothing$. Observe that, for every $v' \in P$, the distance between v' and u in $F[C]$ is either $2\ell - 2$, 2ℓ or $2\ell + 2$ because $F[C]$ is a tree and the distance between v and u is 2ℓ. By construction of $\{C_1, C_2\}$, every vertex at distance $2\ell - 2$ and $2\ell + 2$ from u must belong to C_{3-i}. Thus, every vertex in $P \cap C_i$ is at distance 2ℓ from u. If $\ell = 0$, then we are done because $v = u$ and $P \cap C_i = \varnothing$. Assume that $\ell \neq 0$. As $F[C]$ is a tree, v has only one neighbor w at distance $2\ell - 1$ from u in $F[C]$. Because $F[C]$ is a tree, we deduce that $N_{F[C]}(v) \cap N_{F[C]}(P \cap C_i) = \{w\}$. By definition of X^{2+}, v has at least two neighbors in $C \cap Y$, we conclude that $N_{F[C]}(v) \setminus N_{F[C]}(P \cap C_i) \neq \varnothing$. Hence, we deduce that $\{C_1, C_2\}$ is a good bipartition of $F[C]$.

We deduce that every connected component of F admits a good bipartition and, thus, F admits a good bipartition. Thus, $|X^{2+}| \leqslant 2w$. □

The following lemma generalizes Fact 8 and presents the equivalence between S-forests and forests that we will use in our algorithm.

Lemma 10. *Let $A \subseteq V(G)$, $X \subseteq A$, $\mathcal{P}_X = \mathsf{CC}_G(X \setminus S)$, and $Y \subseteq \overline{A}$. If the graph $G[X \cup Y]$ is an S-forest, then there exists an $\overline{S}$-contraction $\mathcal{P}_Y$ of Y that satisfies the following conditions:*

(1) $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ is a forest,
(2) for all $P \in \mathcal{P}_X \cup \mathcal{P}_Y$ and $v \in (X \cup Y) \cap S$, we have $|N(v) \cap P| \leqslant 1$,
(3) the graph $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ admits a vertex cover VC of size at most $4\mathsf{mim}(A)$ such that the neighborhood of the vertices in VC are pairwise distinct in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$.

Proof. Assume that $G[X \cup Y]$ is an S-forest. Let us explain how we construct $\mathcal{P}_Y$ that satisfies Conditions (1)-(3). First, we initialize $\mathcal{P}_Y = \mathsf{CC}_G(Y \setminus S)$. Observe that there is no cycle in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ that contains a vertex in $\binom{S}{1}$ because $G[X \cup Y]$ is an S-forest. Moreover,

$\mathcal{P}_X$ and $\mathcal{P}_Y$ form two independent sets in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. Consequently, for all the cycles C in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ we have $C = (P_X^1, P_Y^1, P_X^2, P_Y^2, \dots, P_X^t, P_Y^t)$ where $P_X^1, \dots, P_X^t \in \mathcal{P}_X$ and $P_Y^1, \dots, P_Y^t \in \mathcal{P}_Y$. We do the following operation, until the graph $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ is a forest: take a cycle $C = (P_X^1, P_Y^1, P_X^2, P_Y^2, \dots, P_X^t, P_Y^t)$ in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ and replace the blocks $P_Y^1, \dots, P_Y^t$ in $\mathcal{P}_Y$ by the block $P_Y^1 \cup \dots \cup P_Y^t$.

Notice that, whenever we apply the operation on a cycle $C = (P_X^1, P_Y^1, P_X^2, P_Y^2, \dots, P_X^t, P_Y^t)$, there is no vertex $v \in (X \cup Y) \cap S$ such that $\{v\}$ has two neighbors in C, otherwise $G[X \cup Y]$ would not be an S-forest. Thus, Condition (2) is satisfied.

It remains to prove Condition (3). Let VC be the set of vertices of $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ containing:

- vertices that have at least 2 neighbors in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$, and
- vertices $U \in Y_{\downarrow \mathcal{P}_Y}$ for every isolated edge $\{U, T\}$ of $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$.

By construction, it is clear that VC is indeed a vertex cover of $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. We claim that $|\mathsf{VC}| \leqslant 4\mathsf{mim}(A)$. By Fact 7, we know that the size of a maximum matching in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ is at most $\mathsf{mim}(A)$. Let t be the number of isolated edges in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. Observe that the size of a maximum induced matching in the graph obtained from $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ by removing isolated edges is at most $\mathsf{mim}(A) - t$. By Lemma 9, we know that there are at most $4(\mathsf{mim}(A) - t)$ vertices that have at least 2 neighbors in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. We conclude that $|\mathsf{VC}| \leqslant 4\mathsf{mim}(A)$.

Since $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ is a forest, the neighborhood of the vertices that have at least 2 neighbors must be pairwise distinct. We conclude from the construction of VC that the neighborhood of the vertices of VC in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ are pairwise distinct. Hence, Condition (3) is satisfied. $\square$

In the following, we will use Lemma 10 to design some sort of equivalence relation between partial solutions. To this purpose, we use the following notion of indices.

Definition 11 ($\mathbb{I}_x$)**.** *For every $x \in V(T)$, we define the set $\mathbb{I}_x$ of indices as the set of tuples*

$$(X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}) \in 2^{\mathcal{R}^2_{V_x}} \times 2^{\mathcal{R}^1_{V_x}} \times \mathcal{R}^1_{V_x} \times 2^{\mathcal{R}^2_{\overline{V_x}}} \times 2^{\mathcal{R}^1_{\overline{V_x}}}$$

such that $|X_{\mathsf{vc}}^{\overline{S}}| + |X_{\mathsf{vc}}^{S}| + |Y_{\mathsf{vc}}^{\overline{S}}| + |Y_{\mathsf{vc}}^{S}| \leqslant 4\mathsf{mim}(V_x)$.

In the following, we will define partial solutions associated with an index $i \in \mathbb{I}_x$ (a partial solution may be associated with many indices). In order to prove the correctness of our algorithm (the algorithm will not use this concept), we will also define *complement solutions* (the sets $Y \subseteq \overline{V_x}$ and their $\overline{S}$-contractions $\mathcal{P}_Y$) associated with an index i. We will prove that, for every partial solution X and complement solution $(Y, \mathcal{P}_Y)$ associated with i, if the graph $G[X \cup Y]_{\downarrow \mathsf{cc}_G(X \setminus S) \cup \mathcal{P}_Y}$ is a forest, then $G[X \cup Y]$ is an S-forest.

Let us give some intuitions on these indices by explaining how one index is associated with a solution. Let $x \in V(T)$, $X \subseteq V_x$ and $Y \subseteq \overline{V_x}$ such that $G[X \cup Y]$ is an S-forest. Let $\mathcal{P}_X$ and $\mathcal{P}_Y$ be the $\overline{S}$-contractions of X and Y and VC be a vertex cover of $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ given by Lemma 10. Then, X and Y are associated with $i = (X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}) \in \mathbb{I}_x$ such that:

- X_{vc}^{S} (resp. Y_{vc}^{S}) contains the representatives of the blocks $\{v\}$ in VC such that $v \in X \cap S$ (resp. $v \in Y \cap S$) with respect to the 1-neighbor equivalence over V_x (resp. $\overline{V_x}$). We will only use the indices where X_{vc}^{S} contains representatives of singletons, in other words, X_{vc}^{S} is included in $\{\mathsf{rep}^1_{V_x}(\{v\}) \mid v \in V_x\}$ which can be much smaller than $\mathcal{R}^1_{V_x}$. The same observation holds for Y_{vc}^{S}. In Definition 11, we state that X_{vc}^{S} and Y_{vc}^{S} are, respectively, subsets of $2^{\mathcal{R}^1_{V_x}}$ and $2^{\mathcal{R}^1_{\overline{V_x}}}$, for the sake of simplicity.
- $X_{\mathsf{vc}}^{\overline{S}}$ (resp. $Y_{\mathsf{vc}}^{\overline{S}}$) contains the representatives of the blocks in $\mathcal{P}_X \cap \mathsf{VC}$ (resp. $\mathcal{P}_Y \cap \mathsf{VC}$) with respect to the 2-neighbor equivalence relation over V_x (resp. $\overline{V_x}$).
- $X_{\overline{\mathsf{vc}}}$ is the representative of $X \setminus V(\mathsf{VC})$ (the set of vertices which do not belong to the vertex cover) with respect to the 1-neighbor equivalence over V_x.

Because the neighborhood of the blocks in VC are pairwise distinct in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ (Property (3) of Lemma 10), there is a one to one correspondence between the representatives in $X_{\mathsf{vc}}^{\overline{S}} \cup X_{\mathsf{vc}}^{S} \cup Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}$ and the blocks in VC.

While $X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}$ describe VC, the representative set $X_{\overline{\mathsf{vc}}}$ describes the neighborhood of the vertices of X which are not in VC. The purpose of $X_{\overline{\mathsf{vc}}}$ is to make sure that, for every partial solution X and complement solution $(Y, \mathcal{P}_Y)$ associated with i, the set VC described by $X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}$ is a vertex cover of $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. For doing so, it is sufficient to require that $Y \setminus V(\mathsf{VC})$ has no neighbor in $X_{\overline{\mathsf{vc}}}$ for every complement solution $(Y, \mathcal{P}_Y)$ associated with i.

Observe that the sets $X_{\mathsf{vc}}^{\overline{S}}$ and $Y_{\mathsf{vc}}^{\overline{S}}$ contain representatives for the 2-neighbor equivalence. We need the 2-neighbor equivalence to control the S-cycles which might disappear after vertex contractions. To prevent this situation, we require, for example, that every vertex in $X \cap S$ has at most one neighbor in $\overline{R}$ for each $\overline{R} \in Y_{\mathsf{vc}}^{\overline{S}}$. Thanks to the 2-neighbor equivalence, a vertex v in $X \cap S$ has at most one neighbor in $\overline{R} \in Y_{\mathsf{vc}}^{\overline{S}}$ if and only if v has at most one neighbor in the block of $\mathcal{P}_Y$ associated with $\overline{R}$.

In order to define partial solutions associated with i, we need the following notion of auxiliary graph. Given $x \in V(T)$, $X \subseteq V_x$ and $i = (X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}) \in \mathbb{I}_x$, we write $\mathsf{aux}(X, i)$ to denote the graph

$$G[X_{\downarrow \mathsf{cc}_G(X \setminus S)} \mid Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}].$$

Observe that $\mathsf{aux}(X, i)$ is obtained from the graph induced by $X_{\downarrow \mathsf{cc}_G(X \setminus S)} \cup Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}$ by removing the edges between the vertices from $Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}$. Figure 3 illustrates an example of the graph $\mathsf{aux}(X, i)$ and the related notions.

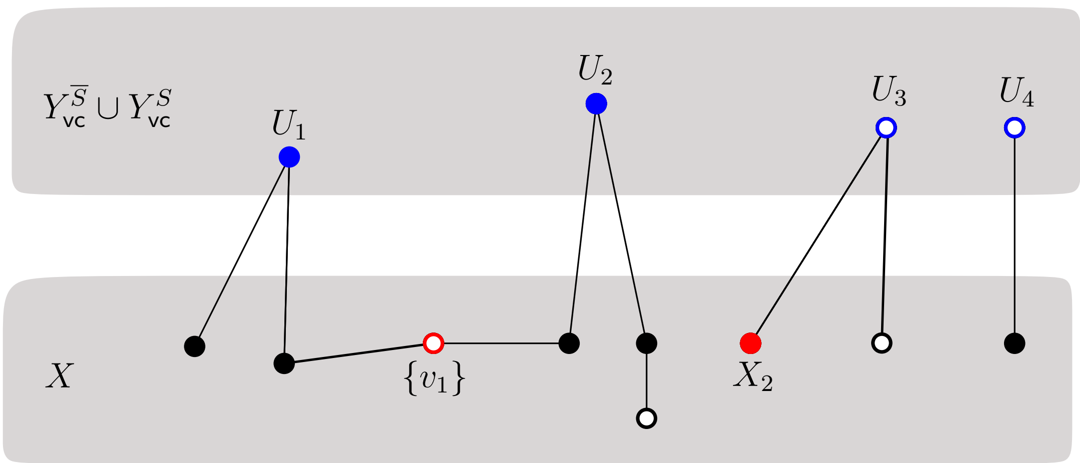


FIGURE 3. An example of a set $X \subseteq V_x$ and its auxiliary graph associated with the index $i = (X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S})$. The white filled vertices correspond to the vertices of S, whereas the blue and red vertices form the set VC. Here, $X_{\mathsf{vc}}^{\overline{S}} = \{R_2\}$ with $R_2 = \mathsf{rep}_{V_x}^2(X_2)$, $X_{\mathsf{vc}}^{S} = \{R_1\}$ with $R_1 = \mathsf{rep}_{V_x}^1(\{v_1\})$, $X_{\overline{\mathsf{vc}}}$ is the representative of the set of black vertices, $Y_{\mathsf{vc}}^{\overline{S}} = \{U_1, U_2\}$, and $Y_{\mathsf{vc}}^{S} = \{U_3, U_4\}$.

We will ensure that, given a complement solution $(Y, \mathcal{P}_Y)$ associated with i, the graph $\mathsf{aux}(X, i)$ is isomorphic to the graph $G[X_{\downarrow \mathsf{cc}_G(X \setminus S)} | (\mathcal{P}_Y \cap \mathsf{VC}) \cup Y_{\mathsf{vc}}^{S}]$. We are now ready to define the notion of partial solution associated with an index i.

Definition 12 (Partial solutions). *Let $x \in V(T)$ and $i = (X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}) \in \mathbb{I}_x$. We say that $X \subseteq V_x$ is a partial solution associated with i if the following conditions are satisfied:*

(a) for every $R \in X_{\mathsf{vc}}^{S}$, there exists a unique $v \in X \cap S$ such that $R \equiv_{V_x}^1 \{v\}$,

(b) for every $R \in X_{\mathsf{vc}}^{\overline{S}}$, there exists a unique $C \in \mathsf{CC}_G(X \setminus S)$ such that $R \equiv_{V_x}^2 C$,

(c) $\mathsf{aux}(X, i)$ is a forest,

(d) for every $C \in \mathsf{CC}_G(X \setminus S)$ and $\{v\} \in Y_{\mathsf{vc}}^{S}$, we have $|N(v) \cap C| \leqslant 1$,

(e) for every $v \in X \cap S$ and $U \in Y_{\mathsf{vc}}^{\overline{S}} \cup \mathsf{CC}_G(X \setminus S)$, we have $|N(v) \cap U| \leqslant 1$,

(f) $X_{\overline{\mathsf{vc}}} \equiv^1_{V_x} X \setminus V(\mathsf{VC}_X)$ *, where* VC_X *contains the blocks* $\{v\} \in \binom{X\cap S}{1}$ *such that* $\mathsf{rep}^1_{V_x}(\{v\}) \in X^S_{\mathsf{vc}}$ *and the components* C *of* $G[X \setminus S]$ *such that* $\mathsf{rep}^2_{V_x}(C) \in X^{\overline{S}}_{\mathsf{vc}}$*.*

Similarly to Definition 12, we define the notion of *complement solutions* associated with an index $i \in \mathbb{I}_x$. We use this concept only to prove the correctness of our algorithm.

Definition 13 (Complement solutions)**.** *Let* $x \in V(T)$ *and* $i = (X^{\overline{S}}_{\mathsf{vc}}, X^S_{\mathsf{vc}}, X_{\overline{\mathsf{vc}}}, Y^{\overline{S}}_{\mathsf{vc}}, Y^S_{\mathsf{vc}}) \in \mathbb{I}_x$*. We call complement solutions associated with* i *all the pairs* $(Y, \mathcal{P}_Y)$ *such that* $Y \subseteq \overline{V_x}$ *and* $\mathcal{P}_Y$ *an* $\overline{S}$*-contraction of* Y *and the following conditions are satisfied:*

(a) for every $U \in Y^S_{\mathsf{vc}}$*, there exists a unique* $v \in Y \cap S$ *such that* $U \equiv^2_{V_x} \{v\}$*,*

(b) for every $U \in Y^{\overline{S}}_{\mathsf{vc}}$*, there exists a unique* $P \in \mathcal{P}_Y$ *such that* $U \equiv^2_{V_x} P$*,*

(c) $G[Y]_{\downarrow \mathcal{P}_Y}$ *is a forest,*

(d) for every $P \in \mathcal{P}_Y$ *and* $\{v\} \in X^S_{\mathsf{vc}}$*, we have* $|N(v) \cap P| \leqslant 1$*,*

(e) for every $y \in Y \cap S$ *and* $R \in X^{\overline{S}}_{\mathsf{vc}} \cup \mathcal{P}_Y$*, we have* $|N(y) \cap R| \leqslant 1$*,*

(f) $N(Y \setminus V(\mathsf{VC}_Y)) \cap X_{\overline{\mathsf{vc}}} = \varnothing$*, where* VC_Y *contains the blocks* $\{v\} \in \binom{Y\cap S}{1}$ *such that* $\mathsf{rep}^1_{\overline{V_x}}(\{v\}) \in Y^S_{\mathsf{vc}}$ *and the blocks* $P \in \mathcal{P}_Y$ *such that* $\mathsf{rep}^2_{\overline{V_x}}(P) \in Y^{\overline{S}}_{\mathsf{vc}}$*.*

Let us give some explanations on the conditions of Definitions 12 and 13. Let $x \in V(T)$, X be a partial solution associated with i, $\mathcal{P}_X = \mathsf{CC}_G(X \setminus S)$, and $(Y, \mathcal{P}_Y)$ be a complement solution associated with i. Conditions (a) and (b) of both definitions guarantee that the set VC described by $X^{\overline{S}}_{\mathsf{vc}}, X^S_{\mathsf{vc}}, Y^{\overline{S}}_{\mathsf{vc}}$ and Y^S_{vc} is included in $X \cup Y$ and that there is an one to one correspondence between the blocks of VC and the representatives in $X^{\overline{S}}_{\mathsf{vc}} \cup X^S_{\mathsf{vc}} \cup Y^{\overline{S}}_{\mathsf{vc}} \cup Y^S_{\mathsf{vc}}$.

Condition (c) of Definition 12 guarantee that the connections between $X_{\downarrow \mathsf{CC}_G(X \setminus S)}$ and VC are acyclic. As explained earlier, Conditions (d) and (e) of both definitions are here to control the S-cycles which might disappear with the vertex contractions.

Finally, as explained earlier, the last conditions of both definitions ensure that VC the set described by $X^{\overline{S}}_{\mathsf{vc}}, X^S_{\mathsf{vc}}, Y^{\overline{S}}_{\mathsf{vc}}$ and Y^S_{vc} is a vertex cover of $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. Notice that $Y_{\overline{\mathsf{vc}}}$ corresponds to the set of vertices of Y which do not belong to a block of VC. Such observations are used to prove the following two results.

Lemma 14. *For every* $X \subseteq V_x$ *and* $Y \subseteq \overline{V_x}$ *such that* $G[X \cup Y]$ *is an* S*-forest, there exist* $i \in \mathbb{I}_x$ *and an* $\overline{S}$*-contraction* $\mathcal{P}_Y$ *of* Y *such that (1)* $G[X \cup Y]_{\downarrow \mathsf{CC}_G(X \setminus S) \cup \mathcal{P}_Y}$ *is a forest, (2)* X *is a partial solution associated with* i *and (3)* $(Y, \mathcal{P}_Y)$ *is a complement solution associated with* i*.*

Proof. Let $X \subseteq V_x$ and $Y \subseteq \overline{V_x}$ such that $G[X \cup Y]$ is an S-forest. By Lemma 10, there exists an $\overline{S}$-contraction $\mathcal{P}_Y$ of Y such that the following properties are satisfied:

(A) $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ is a forest with $\mathcal{P}_X = \mathsf{CC}_G(X \setminus S)$,

(B) for all $P \in \mathcal{P}_X \cup \mathcal{P}_Y$ and all $v \in (X \cup Y) \cap S$, we have $|N(v) \cap P| \leqslant 1$,

(C) the graph $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ admits a vertex cover VC of size at most $4\mathsf{mim}(V_x)$ such that the neighborhood of the vertices in VC are pairwise distinct in $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$.

We construct $i = (X^{\overline{S}}_{\mathsf{vc}}, X^S_{\mathsf{vc}}, X_{\overline{\mathsf{vc}}}, Y^{\overline{S}}_{\mathsf{vc}}, Y^S_{\mathsf{vc}}) \in \mathbb{I}_x$ from VC as follows:

- $X^S_{\mathsf{vc}} = \{\mathsf{rep}^1_{V_x}(\{v\}) \mid \{v\} \in \binom{X\cap S}{1} \cap \mathsf{VC}\}$,
- $X^{\overline{S}}_{\mathsf{vc}} = \{\mathsf{rep}^2_{V_x}(P) \mid P \in \mathcal{P}_X \cap \mathsf{VC}\}$,
- $X_{\overline{\mathsf{vc}}} = \mathsf{rep}^1_{V_x}(X \setminus V(\mathsf{VC}))$,
- $Y^{\overline{S}}_{\mathsf{vc}} = \{\mathsf{rep}^2_{\overline{V_x}}(P) \mid P \in \mathcal{P}_Y \cap \mathsf{VC}\}$,
- $Y^S_{\mathsf{vc}} = \{\mathsf{rep}^1_{\overline{V_x}}(\{v\}) \mid \{v\} \in \binom{Y\cap S}{1} \cap \mathsf{VC}\}$.

Since $|\mathsf{VC}| \leqslant 4\mathsf{mim}(V_x)$, we have $|X^{\overline{S}}_{\mathsf{vc}}| + |X^S_{\mathsf{vc}}| + |Y^{\overline{S}}_{\mathsf{vc}}| + |Y^S_{\mathsf{vc}}| \leqslant 4\mathsf{mim}(V_x)$. By construction of X^S_{vc} and Y^S_{vc}, we deduce that $i \in \mathbb{I}_x$.

We claim that X is a partial solution associated with i . By construction of i, Conditions (a), (b) and (f) of Definition 12 are satisfied. In particular, Condition (a) and (b) follow from the fact that the neighborhood of the blocks in VC are pairwise distinct in $G[X,Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. So, the blocks in $X_{\downarrow \mathcal{P}_X} \cap \mathsf{VC}$ are pairwise non-equivalent for the 1-neighbor equivalence over V_x. Consequently, there is a one to one correspondence between the blocks of $X_{\downarrow \mathcal{P}_X} \cap \mathsf{VC}$ and the representatives in $X^{\overline{S}}_{\mathsf{vc}} \cup X^{S}_{\mathsf{vc}}$.

It remains to prove Conditions (c), (d) and (e). We claim that Condition (c) is satisfied: $\mathsf{aux}(X,i)$ is a forest. Observe that, by construction, $\mathsf{aux}(X,i)$ is isomorphic to the graph $G[X_{\downarrow \mathcal{P}_X} \mid Y_{\downarrow \mathcal{P}_Y} \cap \mathsf{VC}]$. Indeed, for every $P \in Y_{\downarrow \mathcal{P}_Y} \cap \mathsf{VC}$, by construction, there exists a unique $U \in Y^{\overline{S}}_{\mathsf{vc}} \cup Y^{S}_{\mathsf{vc}}$ such that $U \equiv^{1}_{\overline{V_x}} P$ or $U \equiv^{2}_{\overline{V_x}} P$. In both case, we have $N(U) \cap V_x = N(P) \cap V_x$ and thus, the neighborhood of P in $G[X_{\downarrow \mathcal{P}_X} \mid Y_{\downarrow \mathcal{P}_Y} \cap \mathsf{VC}]$ is the same as the neighborhood of U in $\mathsf{aux}(X,i)$. Since $G[X_{\downarrow \mathcal{P}_X} \mid Y_{\downarrow \mathcal{P}_Y} \cap \mathsf{VC}]$ is a subgraph of $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ and this latter graph is a forest, we deduce that $\mathsf{aux}(X,i)$ is also a forest.

We deduce that Conditions (e) and (f) are satisfied from property (B) and Fact 4. Thus X is a partial solution associated with i.

Let us now prove that $(Y, \mathcal{P}_Y)$ is a complement solution associated with i. From the construction of i and with the same arguments used earlier, we can prove that Conditions (a)-(d) of Definition 13 are satisfied. It remains to prove Condition (f): $N(Y \setminus \mathsf{VC}_Y) \cap X_{\overline{\mathsf{vc}}} = \varnothing$. By construction, we have $\mathsf{VC}_Y = V(Y_{\downarrow \mathcal{P}_Y} \cap \mathsf{VC})$. Since, VC is a vertex cover of the graph $G[X,Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$, there are no edges between $X_{\downarrow \mathcal{P}_X} \setminus \mathsf{VC}$ and $Y_{\downarrow \mathcal{P}_Y} \setminus \mathsf{VC}$ in $G[X,Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$. We deduce that $N(Y \setminus \mathsf{VC}_Y) \cap (X \setminus V(\mathsf{VC})) = \varnothing$. Since $X \setminus V(\mathsf{VC}) \equiv^{2}_{V_x} X_{\overline{\mathsf{vc}}}$, we conclude that $N(Y_{\overline{\mathsf{vc}}}) \cap X_{\overline{\mathsf{vc}}} = \varnothing$.

This proves that $(Y \setminus \mathsf{VC}_Y, \mathcal{P}_Y)$ is a complement solution associated with i. $\square$

Lemma 15. *Let $i = (X^{\overline{S}}_{\mathsf{vc}}, X^{S}_{\mathsf{vc}}, X_{\overline{\mathsf{vc}}}, Y^{\overline{S}}_{\mathsf{vc}}, Y^{S}_{\mathsf{vc}}) \in \mathbb{I}_x$, X be a partial solution associated with i, $\mathcal{P}_X = \mathsf{CC}_G(G \setminus S)$, and $(Y, \mathcal{P}_Y)$ be a complement solutions associated with i. If the graph $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ is a forest, then $G[X \cup Y]$ is an S-forest.*

Proof. We prove first that, for all $v \in (X \cup Y) \cap S$ and all $P \in (X \cup Y)_{\mathcal{P}_X \cup \mathcal{P}_Y}$, we have $|N(v) \cap P| \leqslant 1$. Let us prove this statement for a vertex $v \in X \cap S$ the proof is symmetric for $v \in Y \cap S$. Let $P \in (X \cup Y)_{\mathcal{P}_X \cup \mathcal{P}_Y}$. If $P \notin (\mathcal{P}_X \cup \mathcal{P}_Y)$, then P is a singleton and we are done. Condition (e) of Definition 12 guarantees that, for every $C \in \mathcal{P}_X$, we have $|N(v) \cap C| \leqslant 1$.

Assume that $P \in \mathcal{P}_Y$. Suppose first that $\mathsf{rep}^{2}_{\overline{V_x}}(P) \notin Y^{\overline{S}}_{\mathsf{vc}}$. From Condition (e) of Definition 13, we have $N(P) \cap X_{\overline{\mathsf{vc}}} = \varnothing$. Consequently, from the conditions on $X_{\overline{\mathsf{vc}}}$, we deduce that if $\mathsf{rep}^{1}_{V_x}(\{v\}) \notin X^{S}_{\mathsf{vc}}$, then $N(v) \cap P = \varnothing$. If $\mathsf{rep}^{1}_{V_x}(\{v\}) \in X^{S}_{\mathsf{vc}}$, then Condition (c) of Definition 13 ensures that $|N(v) \cap P| \leqslant 1$.

Now, suppose that $\mathsf{rep}^{2}_{\overline{V_x}}(P) \in Y^{\overline{S}}_{\mathsf{vc}}$. By Condition (f) of Definition 12, we know that $|N(v) \cap \mathsf{rep}^{2}_{\overline{V_x}}(P)| \leqslant 1$. From Fact 4, we conclude that $|N(v) \cap P| \leqslant 1$.

We are now ready to prove the statement. Suppose that the graph $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ is a forest. Assume towards a contradiction that there exists an S-cycle C in $G[X \cup Y]$. Let v be a vertex of C which belongs to S and let u and r be the neighbors of v in C.

Let P_u and P_v be the blocks of $(X \cup Y)_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ that contain u and v respectively. We have proved that $|N(v) \cap P| \leqslant 1$ for each $P \in \mathcal{P}_X \cup \mathcal{P}_Y$. Thus, P_u and P_v are two distinct vertices. Since there exists a path between u and v in C that does not contain uv, we deduce that there is a path between P_u and P_v in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ that does not contains the edge $\{u\}\{v\}$. Indeed, this follows from the fact that if there is an edge between two vertices a and b in $G[X \cup Y]$, then either a and b belong to the same block of $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ or there exists an edge between the blocks in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ which contain a and b. We conclude that there exists a cycle in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$, a contradiction. $\square$

For each index $i \in \mathbb{I}_x$, we will design an equivalence relation $\sim_i$ between the partial solutions associated with i. We will prove that, for any partial solutions X and W associated with i, if $X \sim_i W$, then, for any complement solution $Y \subseteq \overline{V_x}$ associated with i, the graph $G[X \cup Y]$ is an S-forest if and only if $G[W \cup Y]$ is an S-forest. Then, given a set of partial solutions $\mathcal{A}$

whose size needs to be reduced, it is sufficient to keep, for each $i \in \mathbb{I}_x$ and each equivalence class $\mathcal{C}$ of $\sim_i$, one partial solution in $\mathcal{C}$ of maximal weight. The resulting set of partial solutions has size bounded by $|\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$ because $\sim_i$ generates at most $(4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$ equivalence classes.

Intuitively, given two partial solutions X and W associated with $i = (X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S})$, we have $X \sim_i W$ if the blocks of VC (i.e., the vertex cover described by i) are *equivalently connected* in $G[X_{\downarrow \mathsf{CC}_G(X\setminus S)} \mid Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}]$ and $G[W_{\downarrow \mathsf{CC}_G(W\setminus S)} \mid Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}]$. In order to compare these connections, we use the following notion of auxiliary graph.

Definition 16 ($\mathsf{CC}(X,i)$). *For each connected component C of $\mathsf{aux}(X,i)$, we denote by C_{vc} the following set:*

- *for every $U \in C$ such that $U \in Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}$, we have $U \in C_{\mathsf{vc}}$,*
- *for every $U = \{v\} \in \binom{X\cap S}{1} \cap C$ and $U \equiv_{V_x}^{1} R$ for some $R \in X_{\mathsf{vc}}^{S}$, we have $R \in C_{\mathsf{vc}}$,*
- *for every $U \in \mathsf{CC}_G(X \setminus S)C$ such that $U \equiv_{V_x}^{2} R$ for some $R \in X_{\mathsf{vc}}^{\overline{S}}$, we have $R \in C_{\mathsf{vc}}$.*

We denote by $\mathsf{CC}(X,i)$ the set $\{C_{\mathsf{vc}} \mid C$ is a connected component of $\mathsf{aux}(X,i)\}$.

For a connected component C of $\mathsf{aux}(X,i)$, the set C_{vc} contains $C \cap (Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S})$ and the representatives of the blocks in $C \cap X_{\downarrow \mathsf{CC}_G(X\setminus S)} \cap \mathsf{VC}$ with VC the vertex cover described by i. Consequently, for every $X \subseteq V_x$ and $i \in \mathbb{I}_x$, the collection $\mathsf{CC}(X,i)$ is partition of $X_{\mathsf{vc}}^{\overline{S}} \cup X_{\mathsf{vc}}^{S} \cup Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}$. For the example given in Figure 3, observe that $\mathsf{CC}(X,i)$ is the partition with the blocks $\{R_1, U_1, U_2\}$, $\{R_2, U_3\}$, and $\{U_4\}$.

Now we are ready to give the notion of equivalence between partial solutions. We say that two partial solutions X, W associated with i are i-equivalent, denoted by $X \sim_i W$, if $\mathsf{CC}(X,i) = \mathsf{CC}(W,i)$. Our next result is the most crucial step. As already explained, our task is to show equivalence between partial solutions under any complement solution with respect to S-forests. Such an equivalence will help us to maintain only a representative set of small size in our algorithm.

Lemma 17. *Let $i \in \mathbb{I}_x$. For every partial solutions X, W associated with i such that $X \sim_i W$ and for every complement solution $(Y, \mathcal{P}_Y)$ associated with i, the graph $G[X \cup Y]_{\downarrow \mathsf{CC}_G(X\setminus S)\cup \mathcal{P}_Y}$ is a forest if and only if the graph $G[W \cup Y]_{\downarrow \mathsf{CC}_G(W\setminus S)\cup \mathcal{P}_Y}$ is a forest.*

Proof. Let X, W be two partial solutions associated with i such that $X \sim_i W$ and let $(Y, \mathcal{P}_Y)$ be a complement solution associated with i. We will use the following notation in this proof. Let $\mathcal{P}_X = \mathsf{CC}_G(X \setminus S)$ and $\mathcal{P}_W = \mathsf{CC}_G(W \setminus S)$.

For $Z \in \{X, W\}$, we denote by VC_Z the set that contains:

- $\{v\}$ for all $v \in Z \cap S$ such that $\mathsf{rep}_{V_x}^{1}(\{v\}) \in X_{\mathsf{vc}}^{S}$,
- all $P \in \mathcal{P}_Z$ such that $\mathsf{rep}_{V_x}^{2}(P) \in X_{\mathsf{vc}}^{\overline{S}}$.

We define also VC_Y as the set that contains:

- $\{v\}$ for all $v \in Y \cap S$ such that $\mathsf{rep}_{\overline{V_x}}^{1}(\{v\}) \in Y_{\mathsf{vc}}^{S}$,
- all $P \in \mathcal{P}_Y$ such that $\mathsf{rep}_{\overline{V_x}}^{2}(P) \in Y_{\mathsf{vc}}^{\overline{S}}$.

The sets $\mathsf{VC}_X, \mathsf{VC}_W$ and VC_Y contain the blocks in $X_{\downarrow \mathcal{P}_X}, W_{\downarrow \mathcal{P}_W}$ and $Y_{\downarrow \mathcal{P}_Y}$, respectively, which belong to the vertex cover described by i.

Finally, for each $Z \in \{X, W\}$, we define the following two subgraphs of $G[Z \cup Y]_{\downarrow \mathcal{P}_Z \cup \mathcal{P}_Y}$:

- $G_Z = G[Z_{\downarrow \mathcal{P}_Z} \mid \mathsf{VC}_Y]$,
- $\overline{G_Z} = G[Z \cup Y]_{\downarrow \mathcal{P}_Z \cup \mathcal{P}_Y} - G_Z$.

As explained in the proof of Lemma 14, for any $Z \in \{X, W\}$, the graph $\mathsf{aux}(Z,i)$ is isomorphic to the graph G_Z. Informally, G_Z contains the edges of $G[Z \cup Y]_{\downarrow \mathcal{P}_Z \cup \mathcal{P}_Y}$ which are induced by $Z_{\downarrow \mathcal{P}_Z}$ and those between $Z_{\downarrow \mathcal{P}_Z}$ and VC_Y. The following fact proves that $\overline{G_Z}$ contains the edges of $G[Z \cup Y]_{\downarrow \mathcal{P}_Z \cup \mathcal{P}_Y}$ that are induced by $Y_{\downarrow \mathcal{P}_Y}$ and those between $Y_{\downarrow \mathcal{P}_Y} \setminus \mathsf{VC}_Y$ and VC_Z.

Fact 18. *For any $Z \in \{X, W\}$, the set $\mathsf{VC}_Z \cup \mathsf{VC}_Y$ is a vertex cover of $G[Z, Y]_{\downarrow \mathcal{P}_Z \cup \mathcal{P}_Y}$.*

Proof. First observe that $N(Y \setminus V(\mathsf{VC}_Y)) \cap X_{\overline{\mathsf{vc}}} = \varnothing$ thanks to Condition (e) of Definition 13. Moreover, we have $X_{\overline{\mathsf{vc}}} \equiv^2_{V_x} Z_{\downarrow \mathcal{P}_Z} \setminus \mathsf{VC}_Z$ by Condition (d) of Definition 12. We conclude that $N(Y_{\downarrow \mathcal{P}_Y} \setminus \mathsf{VC}_Y) \cap (Z_{\downarrow \mathcal{P}_Z} \setminus \mathsf{VC}_Z) = \varnothing$. Hence, $\mathsf{VC}_Z \cup \mathsf{VC}_Y$ is a vertex cover of $G[Z, Y]_{\downarrow \mathcal{P}_Z \cup \mathcal{P}_Y}$. $\square$

Assume that $G[W \cup Y]_{\downarrow \mathcal{P}_W \cup \mathcal{P}_Y}$ contains a cycle C. Our task is to show that $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ contains a cycle as well. We first explore properties of C with respect to G_W and $\overline{G_W}$. Since the graph $\mathsf{aux}(W, i)$ is a forest and it is isomorphic to G_W, we know that C must contain at least one edge from $\overline{G_W}$. Moreover, C must go though a block (i.e., a vertex of $G[W \cup Y]_{\downarrow \mathcal{P}_W \cup \mathcal{P}_Y}$) of $W_{\downarrow \mathcal{P}_W}$ because $G[Y]_{\downarrow \mathcal{P}_Y}$ is a forest. Consequently (and because VC_W is a vertex cover of $G[W \cup Y]_{\downarrow \mathcal{P}_W \cup \mathcal{P}_Y}$), we deduce that

- C is the concatenation of non-empty edge-disjoint *paths* P between blocks in $\mathsf{VC}_W \cup \mathsf{VC}_Y$ such that P is a path of G_W or $\overline{G_W}$.

At least one of these paths is in $\overline{G_W}$ and, potentially, C may be entirely contained in $\overline{G_W}$. Figure 4 presents two possible interactions between C and the graphs G_W and $\overline{G_W}$.

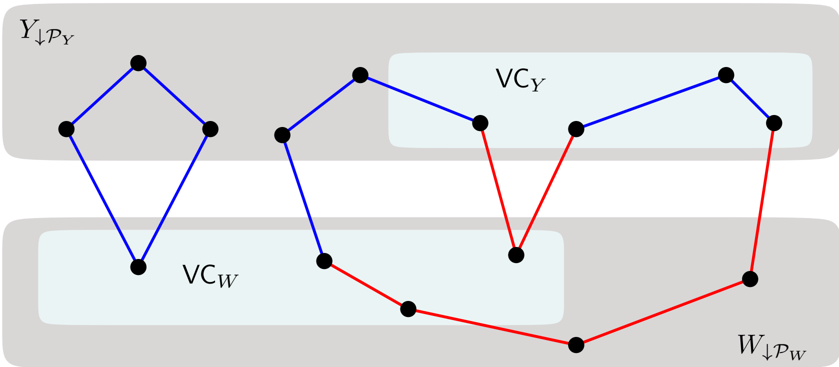


FIGURE 4. How cycles in $G[W \cup Y]_{\downarrow \mathcal{P}_W \cup \mathcal{P}_Y}$ may interact with the graphs G_W and $\overline{G_W}$. The red edges belong to G_W and the blue edges belong to $\overline{G_W}$.

Let P be a non-empty edge-disjoint path between two blocks U and T of $\mathsf{VC}_W \cup \mathsf{VC}_Y$ in $G[W \cup Y]_{\downarrow \mathcal{P}_W \cup \mathcal{P}_Y}$. In order to construct a corresponding non-empty edge-disjoint path in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$, we will make use of the fact that both partial solutions X, W are associated with $i = (X^{\overline{S}}_{\mathsf{vc}}, X^{S}_{\mathsf{vc}}, X_{\overline{\mathsf{vc}}}, Y^{\overline{S}}_{\mathsf{vc}}, Y^{S}_{\mathsf{vc}}) \in \mathbb{I}_x$. In particular, we define U_X and U_i as the analogs of U in $\mathsf{VC}_X \cup \mathsf{VC}_Y$ and $X^{\overline{S}}_{\mathsf{vc}} \cup X^{S}_{\mathsf{vc}} \cup Y^{\overline{S}}_{\mathsf{vc}} \cup Y^{S}_{\mathsf{vc}}$, respectively, as follows (recall that $U \in \mathsf{VC}_W \cup \mathsf{VC}_Y$):

- if $U \in \mathcal{P}_W$, then U_X (resp. U_i) is the unique element of $\mathcal{P}_X$ (resp. $X^{\overline{S}}_{\mathsf{vc}}$) such that $U \equiv^2_{V_x} U_X$ (resp. $U \equiv^2_{V_x} U_i$),
- if $U = \{v\} \in \binom{W \cap S}{1}$, then U_X and U_i are the unique elements in $\binom{X \cap S}{1}$ and X^{S}_{vc}, respectively, such that $U \equiv^1_{V_x} U_X \equiv^1_{V_x} U_i$,
- if $U \in \mathcal{P}_Y$, then $U_X = U$ and U_i is the unique element of $Y^{\overline{S}}_{\mathsf{vc}}$ such that $U \equiv^2_{\overline{V_x}} U_i$;
- otherwise, if $U = \{v\} \in \binom{Y \cap S}{1}$, then $U_X = U$ and U_i is the unique element of Y^{S}_{vc} such that $U \equiv^1_{\overline{V_x}} U_i$.

Similarly, we define T_X and T_i as the analogs of T. Observe that U_X, T_X, U_i and T_i exist by Conditions (a) and (b) of Definition 12 and Definition 13. We will prove that if P is a path in G_W (resp. $\overline{G_W}$), then there exists a non-empty path in G_X (resp. $\overline{G_X}$) between U_X and T_X. This is sufficient to prove the claim because thanks to this, we can construct from C a cycle in $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ by concatenating the corresponding paths.

First, assume that P is a path of G_W. Observe that U_i and T_i belong to the same partition class of $\mathsf{CC}(W, i)$. This follows from the definitions of U_i, T_i and the fact that G_W is isomorphic to $\mathsf{aux}(W,i)$. As $W \sim_i X$, we deduce that U_i and T_i belong to the same partition class of $\mathsf{CC}(X,i)$. By construction, U_i and T_i are the analogs of U_X and T_X in $X_{\mathsf{vc}}^{\overline{S}} \cup X_{\mathsf{vc}}^{S} \cup Y_{\mathsf{vc}}^{\overline{S}} \cup Y_{\mathsf{vc}}^{S}$. We conclude that U_X and T_X are connected in G_X via a path P'. We claim that P' is not empty because we have $U_X \neq T_X$. As P is a non empty path of G_W and because G_W is acyclic, we know that U and T are distinct. Hence, by the construction of U_X and T_X, we deduce that $U_X \neq T_X$.

Now, assume that P is a path of $\overline{G_W}$. We further decompose P into a concatenation of non-empty subpaths such that the internal vertices (i.e., the internal blocks[1]) of each subpath do not belong to $\mathsf{VC}_W \cup \mathsf{VC}_Y$. Our aim is to prove that for every path of $\overline{G_W}$ in which no internal vertex belongs to $\mathsf{VC}_W \cup \mathsf{VC}_Y$, there is a corresponding non-empty path in $\overline{G_X}$. Observe that such an argument implies the existence of a general path in $\overline{G_X}$ by taking the concatenation of every such subpath. Thus, we assume that the internal vertices of P do not belong to $\mathsf{VC}_W \cup \mathsf{VC}_Y$.

By assumption, we show that all internal vertices of P belong to $Y_{\downarrow \mathcal{P}_Y}$. Indeed, since $\mathsf{VC}_W \cup \mathsf{VC}_Y$ is a vertex cover of $G[W,Y]_{\downarrow \mathcal{P}_W \cup \mathcal{P}_Y}$, there are no edges incident to $W_{\downarrow \mathcal{P}_W} \setminus VC_W$ in $\overline{G_W}$. So, the vertices of P which belong to $W_{\downarrow \mathcal{P}_W}$ are in VC_W. As P has no internal vertices in $\mathsf{VC}_W \cup \mathsf{VC}_Y$, the internal vertices of P belong to $Y_{\downarrow \mathcal{P}_Y}$.

If both endpoints of P belong to $Y_{\downarrow \mathcal{P}_Y}$, then the claim follows because $U_X = U$, $T_X = T$, and P is a non-empty path of $G[Y]_{\downarrow \mathcal{P}_Y}$ which is a subgraph of $\overline{G_X}$.

Suppose that U belongs to VC_W. Let $Q \in Y_{\downarrow \mathcal{P}_Y}$ be the neighbor of U in P. Observe that Q exists in $\overline{G_X}$ because all blocks of $\mathcal{P}_Y$ are contained in $\overline{G_X}$ by construction. We claim that Q is adjacent to U_X in $\overline{G_X}$. By definition of U_X, we have $U \equiv^2_{V_x} U_X$ and in particular $N(U) \cap \overline{V_x} = N(U_X) \cap \overline{V_x}$. As U and Q are adjacent in $\overline{G_W}$, we deduce that $N(U) \cap Q \neq \varnothing$. It follows that $N(U_X) \cap Q \neq \varnothing$ and thus Q and U_X are adjacent in $\overline{G_X}$. Symmetrically, we can prove that if $T \in W_{\downarrow \mathcal{P}_W}$, then the neighbor of T in P is adjacent to T_X in $\overline{G_X}$. We conclude that there exists a path P' between U_X and T_X in $\overline{G_X}$.

Let us show that P' is indeed a non-empty path. If $U_X \neq T_X$, then P' is obviously not empty. Suppose that $U_X = T_X$. By construction of U_X and T_X, we deduce that $U = T$. As P is a non-empty path, $U = T$ implies that P is a cycle. We deduce that both U and U_X belong to $W_{\downarrow \mathcal{P}_W}$ because $G[Y]_{\downarrow \mathcal{P}_Y}$ is acyclic and, by assumption, all internal vertices of P belong to $Y_{\downarrow \mathcal{P}_Y}$. Since there is no edge between the vertices of $W_{\downarrow \mathcal{P}_W}$ in $\overline{G_W}$, the path P contains at least one edge of $G[Y]_{\downarrow \mathcal{P}_Y}$. By construction, P' contains all the edges of P which belong to $G[Y]_{\downarrow \mathcal{P}_Y}$. Therefore, we conclude that P' is not empty. ☐

The following theorem proves that, for every set of partial solutions $\mathcal{A} \subseteq 2^{V_x}$, we can compute a small subset $\mathcal{B} \subseteq \mathcal{A}$ such that $\mathcal{B}$ *represents* $\mathcal{A}$, i.e., for every $Y \subseteq \overline{V_x}$, the best solutions we obtain from the union of Y with a set in $\mathcal{A}$ is as good as the ones we obtain from $\mathcal{B}$. Firstly, we formalize this notion of representativity.

Definition 19 (Representativity)**.** *Let $x \in V(T)$. For every $\mathcal{A} \subseteq 2^{V_x}$ and $Y \subseteq \overline{V_x}$, we define*

$$\mathsf{best}(\mathcal{A}, Y) = \max\{\mathsf{w}(X) \mid X \in \mathcal{A} \text{ and } G[X \cup Y] \text{ is an } S\text{-forest}\}.$$

Given $\mathcal{A}, \mathcal{B} \subseteq 2^{V_x}$, we say that $\mathcal{B}$ represents *$\mathcal{A}$ if, for every $Y \subseteq \overline{V_x}$, we have $\mathsf{best}(\mathcal{A}, Y) = \mathsf{best}(\mathcal{B}, Y)$.*

Theorem 20. *Let $x \in V(T)$. Then, there exists an algorithm* reduce *that, given a set $\mathcal{A} \subseteq 2^{V_x}$, outputs in time $O(|\mathcal{A}| \cdot |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)} \cdot \mathsf{s\text{-}nec}_2(V_x) \cdot n^3)$ a subset $\mathcal{B} \subseteq \mathcal{A}$ such that $\mathcal{B}$ represents $\mathcal{A}$ and $|\mathcal{B}| \leqslant |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$.*

Proof. Given $\mathcal{A} \subseteq 2^{V_x}$ and $i \in \mathbb{I}_x$, we define $\mathsf{reduce}(\mathcal{A}, i)$ as the operation which returns a set containing one partial solution $X \in \mathcal{A}$ associated with i of each equivalence class of $\sim_i$ such that $\mathsf{w}(X)$ is maximum. Moreover, we define $\mathsf{reduce}(\mathcal{A}) = \bigcup_{i \in \mathbb{I}_x} \mathsf{reduce}(\mathcal{A}, i)$.

[1]For the rest of the proof, we use the term *vertices* to refer to the *blocks* of contracted graphs in order to use the standard terminology related to paths and cycles.

We prove first that $\mathsf{reduce}(\mathcal{A})$ represents $\mathcal{A}$, that is $\mathsf{best}(\mathcal{A}, Y) = \mathsf{best}(\mathsf{reduce}(\mathcal{A}), Y)$ for all $Y \subseteq \overline{V_x}$. Let $Y \subseteq \overline{V_x}$. Since $\mathsf{reduce}(\mathcal{A}) \subseteq \mathcal{A}$, we already have $\mathsf{best}(\mathsf{reduce}(\mathcal{A}), Y) \leqslant \mathsf{best}(\mathcal{A}, Y)$. Consequently, if there is no $X \in \mathcal{A}$ such that $G[X \cup Y]$ is an S-forest, we have $\mathsf{best}(\mathsf{reduce}(\mathcal{A}), Y) = \mathsf{best}(\mathcal{A}, Y) = -\infty$.

Assume that there exists $X \in \mathcal{A}$ such that $G[X \cup Y]$ is an S-forest. Let $X \in \mathcal{A}$ such that $G[X \cup Y]$ is an S-forest and $\mathsf{w}(X) = \mathsf{best}(\mathcal{A}, Y)$. By Lemma 14, there exists $i \in \mathbb{I}_x$ and an $\overline{S}$-contraction $\mathcal{P}_Y$ of Y such that (1) $G[X \cup Y]_{\downarrow \mathsf{CC}_{G}(X \setminus S) \cup \mathcal{P}_Y}$ is a forest, (2) X is a partial solution associated with i and (3) $(Y, \mathcal{P}_Y)$ is a complement solution associated with i.

From the construction of $\mathsf{reduce}(\mathcal{A}, i)$, there exists $W \in \mathsf{reduce}(\mathcal{A})$ such that W is a partial solution associated with i, $X \sim_i W$, and $\mathsf{w}(W) \geq \mathsf{w}(X)$. By Lemma 17 and since $G[X \cup Y]_{\downarrow \mathsf{CC}_{G}(X \setminus S) \cup \mathcal{P}_Y}$ is a forest, we deduce that $G[W \cup Y]_{\downarrow \mathsf{CC}_{G}(W \setminus S) \cup \mathcal{P}_Y}$ is a forest too. Thanks to Lemma 15, we deduce that $G[W \cup Y]$ is an S-forest. As $\mathsf{w}(W) \geq \mathsf{w}(X) = \mathsf{best}(\mathcal{A}, Y)$, we conclude that $\mathsf{best}(\mathcal{A}, Y) = \mathsf{best}(\mathsf{reduce}(\mathcal{A}), Y)$. Hence, $\mathsf{reduce}(\mathcal{A})$ represents $\mathcal{A}$.

We claim that $|\mathsf{reduce}(\mathcal{A})| \leqslant |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$. For every $i = (X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}) \in \mathbb{I}_x$ and $X \subseteq V_x$, by definition, $\mathsf{CC}(X, i)$ is a partition of $X_{\mathsf{vc}}^{\overline{S}}, X_{\mathsf{vc}}^{S}, X_{\overline{\mathsf{vc}}}, Y_{\mathsf{vc}}^{\overline{S}}, Y_{\mathsf{vc}}^{S}$. Since $|X_{\mathsf{vc}}^{\overline{S}}| + |X_{\mathsf{vc}}^{S}| + |Y_{\mathsf{vc}}^{\overline{S}}| + |Y_{\mathsf{vc}}^{S}| \leqslant 4\mathsf{mim}(V_x)$, there are at most $(4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$ possible partitions for $\mathsf{CC}(X, i)$. We deduce that, for every $i \in \mathbb{I}_x$, the relation $\sim_i$ generates at most $(4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$ equivalence classes, so $|\mathsf{reduce}(\mathcal{A}, i)| \leqslant (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$ for every $i \in \mathbb{I}_x$. By construction, we conclude that $|\mathsf{reduce}(\mathcal{A})| \leqslant |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$.

It remains to prove the runtime. By Lemma 6, we can compute in time $O(\mathsf{s\text{-}nec}_2(V_x) \cdot n^2 \cdot \log(\mathsf{s\text{-}nec}_2(V_x)))$ the sets $\mathcal{R}^1_{V_x}, \mathcal{R}^2_{V_x}, \mathcal{R}^2_{\overline{V_x}}$ and a data structure which computes $\mathsf{rep}^1_{V_x}, \mathsf{rep}^2_{V_x}$ and $\mathsf{rep}^2_{\overline{V_x}}$ in time $O(n^2 \cdot \log(\mathsf{s\text{-}nec}_2(V_x)))$; notice that we implicitly use the fact that $\mathsf{nec}_1(A) \leqslant \mathsf{nec}_2(A)$ for all $A \subseteq V(G)$.

Given $\mathcal{R}^1_{V_x}, \mathcal{R}^2_{V_x}, \mathcal{R}^2_{\overline{V_x}}$, we can compute $\mathbb{I}_x$ in time $O(|\mathbb{I}_x|)$. Moreover, for $i \in \mathbb{I}_x$ and $X \subseteq V_x$, we can compute $\mathsf{aux}(X, i)$, $\mathsf{CC}(X, i)$ and decide whether X is a partial solution associated with i in time $O(n^3 \cdot \mathsf{s\text{-}nec}_2(V_x))$. Consequently, given two partial solutions X, W associated with i, we can decide whether $X \sim_i W$ in time $O(n^3 \cdot \mathsf{s\text{-}nec}_2(V_x))$.

Since $|\mathsf{reduce}(\mathcal{A})| \leqslant |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$, we conclude that $\mathsf{reduce}(\mathcal{A})$ is computable in time $O(|\mathcal{A}| \cdot |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)} \cdot \mathsf{s\text{-}nec}_2(V_x) \cdot n^3)$. $\square$

We are now ready to prove the main theorem of this paper. For two subsets $\mathcal{A}$ and $\mathcal{B}$ of $2^{V(G)}$, we define the *merging* of $\mathcal{A}$ and $\mathcal{B}$, denoted by $\mathcal{A} \bigotimes \mathcal{B}$, as

$$\mathcal{A} \bigotimes \mathcal{B} = \begin{cases} \varnothing & \text{if } \mathcal{A} = \varnothing \text{ or } \mathcal{B} = \varnothing, \\ \{X \cup Y \mid X \in \mathcal{A} \text{ and } Y \in \mathcal{B}\} & \text{otherwise.} \end{cases}$$

Theorem 21. *There exists an algorithm that, given an n-vertex graph G and a rooted layout (T, δ) of G, solves* Subset Feedback Vertex Set *in time*

$$\sum_{x \in V(T)} |\mathbb{I}_x|^3 \cdot (4\mathsf{mim}(V_x))^{12\mathsf{mim}(V_x)} \cdot \mathsf{s\text{-}nec}_2(V_x) \cdot n^3.$$

Proof. The algorithm is a usual bottom-up dynamic programming algorithm. For every node x of T, the algorithm computes a set of partial solutions $\mathcal{A}_x \subseteq 2^{V_x}$ such that $\mathcal{A}_x$ represents 2^{V_x} and $|\mathcal{A}_x| \leqslant |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$. For the leaves x of T such that $V_x = \{v\}$, we simply take $\mathcal{A}_x = 2^{V_x} = \{\varnothing, \{v\}\}$. In order to compute $\mathcal{A}_x$ for x an internal node of T with a and b as children, our algorithm will simply compute $\mathcal{A}_x = \mathsf{reduce}(\mathcal{A}_a \bigotimes \mathcal{A}_b)$.

By Theorem 20, we have $|\mathcal{A}_x| \leqslant |\mathbb{I}_x| \cdot (4\mathsf{mim}(V_x))^{4\mathsf{mim}(V_x)}$, for every node x of T. The following claim helps us to prove that $\mathcal{A}_x$ represents 2^{V_x} for the internal nodes x of T.

Claim 22. *Let x be an internal of T with a and b as children. If $\mathcal{A}_a$ and $\mathcal{A}_b$ represent, respectively, 2^{V_a} and 2^{V_b}, then $\mathcal{A}_a \bigotimes \mathcal{A}_b$ represents 2^{V_x}.*

Proof. Assume that $\mathcal{A}_a$ and $\mathcal{A}_b$ represent, respectively, 2^{V_a} and 2^{V_b}. We have to prove that, for every $Y \subseteq \overline{V_x}$, we have $\mathsf{best}(\mathcal{A}_a \bigotimes \mathcal{A}_b, Y) = \mathsf{best}(2^{V_x}, Y)$. Let $Y \subseteq \overline{V_x}$. By definition of best, we have the following

$$\begin{aligned}\mathsf{best}(\mathcal{A}_a \bigotimes \mathcal{A}_b, Y) &= \mathsf{max}\{\mathsf{w}(X) + \mathsf{w}(W) \mid X \in \mathcal{A}_a \wedge W \in \mathcal{A}_b \wedge G[X \cup W \cup Y] \text{ is an } S\text{-forest}\} \\ &= \mathsf{max}\{\mathsf{best}(\mathcal{A}_a, W \cup Y) + \mathsf{w}(W) \mid W \in \mathcal{A}_b\}.\end{aligned}$$

As $\mathcal{A}_a$ represents 2^{V_a}, we have

$$\begin{aligned}\mathsf{best}(\mathcal{A}_a \bigotimes \mathcal{A}_b, Y) &= \mathsf{max}\{\mathsf{best}(2^{V_a}, W \cup Y) + \mathsf{w}(W) \mid W \in \mathcal{A}_b\}. \\ &= \mathsf{max}\{\mathsf{w}(X) + \mathsf{w}(W) \mid X \in 2^{V_a} \wedge W \in \mathcal{A}_b \wedge G[X \cup W \cup Y] \text{ is an } S\text{-forest}\}\end{aligned}$$

With the same arguments and since $\mathcal{A}_b$ represents 2^{V_b}, we infer the following

$$\begin{aligned}\mathsf{best}(\mathcal{A}_a \bigotimes \mathcal{A}_b, Y) &= \mathsf{max}\{\mathsf{best}(\mathcal{A}_b, X \cup Y) + \mathsf{w}(X) \mid X \in 2^{V_a}\} \\ &= \mathsf{max}\{\mathsf{w}(X) + \mathsf{w}(W) \mid X \in 2^{V_a} \wedge W \in 2^{V_b} \wedge G[X \cup W \cup Y] \text{ is an } S\text{-forest}\} \\ &= \mathsf{best}(2^{V_a} \bigotimes 2^{V_b}, Y).\end{aligned}$$

We conclude that $\mathsf{best}(\mathcal{A}_a \bigotimes \mathcal{A}_b, Y)$ equals $\mathsf{best}(2^{V_x}, Y)$. Hence, $\mathcal{A}_a \bigotimes \mathcal{A}_b$ represents 2^{V_x}. □

For the leaves x of T, we obviously have that $\mathcal{A}_x$ represents 2^{V_x}, since $\mathcal{A}_x = 2^{V_x}$. From Claim 22 and by induction, we deduce that $\mathcal{A}_x$ represents 2^{V_x} for every node x of T. Let r be the root of T. As $\mathcal{A}_r$ represents $2^{V(G)}$, by Definition 19, $\mathcal{A}_r$ contains a set X of maximum size such that $G[X]$ is an S-forest.

It remains to prove the running time. Observe that, for every internal node x of T with a and b as children, the size of $\mathcal{A}_a \bigotimes \mathcal{A}_b$ is at most $|\mathbb{I}_x|^2 \cdot (4\mathsf{mim}(V_x))^{8\mathsf{mim}(V_x)}$. By Theorem 20, the set $\mathcal{A}_x = \mathsf{reduce}(\mathcal{A}_a \bigotimes \mathcal{A}_b)$ is computable in time $|\mathbb{I}_x|^3 \cdot (4\mathsf{mim}(V_x))^{12\mathsf{mim}(V_x)} \cdot \mathsf{s\text{-}nec}_2(V_x) \cdot n^3$. This proves the running time. □

3.1. **Algorithmic consequences.** In order to obtain the algorithmic consequences of our meta-algorithm given in Theorem 21, we need the following lemma which bounds the size of each table index with respect to the considered parameters.

Lemma 23. *For every $x \in V(T)$, the size of $\mathbb{I}_x$ is upper bounded by:*

- $2^{O(\mathsf{rw}(V_x)^3)}$,
- $\mathsf{rw}_{\mathbb{Q}}(V_x)^{O(\mathsf{rw}_{\mathbb{Q}}(V_x)^2)}$,
- $n^{O(\mathsf{mim}(V_x)^2)}$.

Proof. For $A \subseteq V(G)$, let $\mathsf{mw}(A)$ be the number of different rows in the matrix $M_{A,\overline{A}}$. Observe that, for every $A \subseteq V(G)$, we have $\mathsf{mw}(A) = \{\mathsf{rep}^1_A(\{v\}) \mid v \in A\}$. From Definition 11, we have $|X^{\overline{S}}_{\mathsf{vc}}| + |X^S_{\mathsf{vc}}| + |Y^{\overline{S}}_{\mathsf{vc}}| + |Y^S_{\mathsf{vc}}| \leqslant 4\mathsf{mim}(V_x)$, for every $(X^{\overline{S}}_{\mathsf{vc}}, X^S_{\mathsf{vc}}, X_{\overline{\mathsf{vc}}}, Y^{\overline{S}}_{\mathsf{vc}}, Y^S_{\mathsf{vc}}) \in \mathbb{I}_x$. Thus, the size of $\mathbb{I}_x$ is at most

$$\mathsf{nec}_1(V_x) \cdot \big(\mathsf{nec}_2(V_x) + \mathsf{mw}(V_x) + \mathsf{nec}_2(\overline{V_x}) + \mathsf{mw}(\overline{V_x})\big)^{4\mathsf{mim}(V_x)}.$$

Rank-width. By Lemma 5, we have $\mathsf{nec}_1(V_x) \leqslant 2^{\mathsf{rw}(V_x)^2}$, $\mathsf{nec}_2(V_x), \mathsf{nec}_2(\overline{V_x}) \leqslant 2^{2\mathsf{rw}(V_x)^2}$. Moreover, there is at most $2^{\mathsf{rw}(V_x)}$ different rows in the matrices $M_{V_x,\overline{V_x}}$ and $M_{\overline{V_x},V_x}$, so $\mathsf{mw}(V_x)$ and $\mathsf{mw}(\overline{V_x})$ are upper bounded by $2^{\mathsf{rw}(V_x)}$. By Lemma 3, we have $4\mathsf{mim}(V_x) \leqslant 4\mathsf{rw}(V_x)$. We deduce from these inequalities that $|\mathbb{I}_x| \leqslant 2^{\mathsf{rw}(V_x)^2} \cdot (2^{2\mathsf{rw}(V_x)^2+1} + 2^{\mathsf{rw}(V_x)})^{4\mathsf{rw}(V_x)} \in 2^{O(\mathsf{rw}(V_x)^3)}$.

$\mathbb{Q}$-rank-width. By Lemma 5, we have $\mathsf{nec}_1(V_x) \leqslant (\mathsf{rw}_{\mathbb{Q}}(V_x)+1)^{\mathsf{rw}_{\mathbb{Q}}(V_x)}$, $\mathsf{nec}_2(V_x), \mathsf{nec}_2(\overline{V_x}) \leqslant (2\mathsf{rw}_{\mathbb{Q}}(V_x)+1)^{\mathsf{rw}_{\mathbb{Q}}(V_x)}$. Moreover, there is at most $2^{\mathsf{rw}_{\mathbb{Q}}(V_x)}$ different rows in the matrices $M_{V_x,\overline{V_x}}$ and $M_{\overline{V_x},V_x}$, so $\mathsf{mw}(V_x)$ and $\mathsf{mw}(\overline{V_x})$ are upper bounded by $2^{\mathsf{rw}_{\mathbb{Q}}(V_x)}$. By Lemma 3, we have $4\mathsf{mim}(V_x) \leqslant 4\mathsf{rw}_{\mathbb{Q}}(V_x)$. We deduce from these inequalities that

$$|\mathbb{I}_x| \leqslant (\mathsf{rw}_{\mathbb{Q}}(V_x)+1)^{\mathsf{rw}_{\mathbb{Q}}(V_x)} \cdot \left(2 \cdot (2\mathsf{rw}_{\mathbb{Q}}(V_x)+1)^{\mathsf{rw}_{\mathbb{Q}}(V_x)} + 2^{\mathsf{rw}_{\mathbb{Q}}(V_x)+1}\right)^{4\mathsf{rw}_{\mathbb{Q}}(V_x)} \in \mathsf{rw}_{\mathbb{Q}}(V_x)^{O(\mathsf{rw}_{\mathbb{Q}}(V_x)^2)}.$$

Mim-width. By Lemma 5, we know that $\mathsf{nec}_1(V_x) \leqslant |V_x|^{\mathsf{mim}(V_x)}$, $\mathsf{nec}_2(V_x) \leqslant |V_x|^{2\mathsf{mim}(V_x)}$, and $\mathsf{nec}_2(\overline{V_x}) \leqslant |\overline{V_x}|^{2\mathsf{mim}(V_x)}$. We can assume that $n > 2$ (otherwise the problem is trivial), so $\mathsf{nec}_2(V_x) + \mathsf{nec}_2(\overline{V_x}) \leqslant |V_x|^{2\mathsf{mim}(V_x)} + |V_x|^{2\mathsf{mim}(V_x)} \leqslant n^{\mathsf{mim}(V_x)}$. Moreover, notice that, for every $A \subseteq V(G)$, we have $\{\mathsf{rep}^1_A(\{v\}) \mid v \in A\} \leqslant |A|$.

We deduce that $|\mathbb{I}_x| \leqslant n^{\mathsf{mim}(V_x)} \cdot (n + n^{2\mathsf{mim}(V_x)})^{4\mathsf{mim}(V_x)}$. As we assume that $n > 2$, we have $|\mathbb{I}_x| \leqslant n^{8\mathsf{mim}(V_x)^2+5\mathsf{mim}(V_x)} \in n^{O(\mathsf{mim}(V_x)^2)}$.

□

Now we are ready to state our algorithms with respect to the parameters rank-width $\mathsf{rw}(G)$ and $\mathbb{Q}$-rank-width $\mathsf{rw}_{\mathbb{Q}}(G)$. In particular, with our next result we show that Subset Feedback Vertex Set is in FPT parameterized by $\mathsf{rw}_{\mathbb{Q}}(G)$ or $\mathsf{rw}(G)$.

Theorem 24. *There exist algorithms that solve* Subset Feedback Vertex Set *in time* $2^{O(\mathsf{rw}(G)^3)} \cdot n^4$ *and* $\mathsf{rw}_{\mathbb{Q}}(G)^{O(\mathsf{rw}_{\mathbb{Q}}(G)^2))} \cdot n^4$.

Proof. We first compute a rooted layout $\mathcal{L} = (T, \delta)$ of G such that $\mathsf{rw}(\mathcal{L}) \in O(\mathsf{rw}(G))$ and $\mathsf{rw}_{\mathbb{Q}}(\mathcal{L}) \in O(\mathsf{rw}_{\mathbb{Q}}(G))$. This is achieved through a $(3k+1)$-approximation algorithm that runs in FPT time $O(8^k \cdot n^4)$ parameterized by $k \in \{\mathsf{rw}(G), \mathsf{rw}_{\mathbb{Q}}(G)\}$ [32]. Then, we apply the algorithm given in Theorem 21. Observe that for every node $x \in V(T)$, by Lemma 23 and Lemma 5, we have $|\mathbb{I}_x|^3 \cdot \mathsf{s\text{-}nec}_2(V_x) \in 2^{O(\mathsf{rw}(V_x)^3)}$ and $|\mathbb{I}_x|^3 \cdot \mathsf{s\text{-}nec}_2(V_x) \in \mathsf{rw}_{\mathbb{Q}}(G)^{O(\mathsf{rw}_{\mathbb{Q}}(G)^2))}$. Moreover, Lemma 3 implies that $\mathsf{mim}(G)^{\mathsf{mim}(G)} \leqslant 2^{\mathsf{rw}(G)\cdot\log(\mathsf{rw}(G))}$ and $\mathsf{mim}(G)^{\mathsf{mim}(G)} \leqslant \mathsf{rw}_{\mathbb{Q}}(G)^{\mathsf{rw}_{\mathbb{Q}}(G)}$. Therefore, we get the claimed runtimes for SFVS since T contains at most n nodes. As explained earlier NMC can then also be solved in the same runtime. □

Regarding mim-width, our algorithm given below shows that Subset Feedback Vertex Set is in XP parameterized by $\mathsf{mim}(G)$ if given an optimal rooted layout. Note that we can hardly avoid the dependence of the exponent for $\mathsf{mim}(G)$ in the claimed running time, since Subset Feedback Vertex Set is known to be $\mathsf{W}[1]$-hard when parameterized by $\mathsf{mim}(G)$ even for the special case of $S = V(G)$ [27]. Moreover, contrary to the algorithms given in Theorem 24, here we need to assume that the input graph is given with a rooted layout. However, our next result actually provides a unified polynomial-time algorithm for Subset Feedback Vertex Set on well-known graph classes having bounded mim-width and for which a layout of bounded mim-width can be computed in polynomial time [1] (e.g., interval graphs, circular arc graphs, permutation graphs, Dilworth-k graphs and k-polygon graphs for all fixed k).

Theorem 25. *There exists an algorithm that, given an n-vertex graph G and a rooted layout $\mathcal{L}$ of G, solves* Subset Feedback Vertex Set *in time* $n^{O(\mathsf{mim}(\mathcal{L})^2)}$.

Proof. We apply the algorithm given in Theorem 21. By Lemmas 23 and 5, we have $|\mathbb{I}_x|^3 \cdot \mathsf{s\text{-}nec}_2(V_x) \in n^{O(\mathsf{mim}(V_x)^2)}$. Therefore, the claimed runtime for SFVS follows by the fact that the rooted tree T of $\mathcal{L}$ contains at most n nodes. As explained earlier NMC can then also be solved in the same runtime. □

Let us relate our results for Subset Feedback Vertex Set to the Node Multiway Cut. It is known that Node Multiway Cut reduces to Subset Feedback Vertex Set [19]. In fact, we can solve Node Multiway Cut by adding a single S-vertex with a large weight that is adjacent to all terminals and, then, run our algorithms for Subset Feedback Vertex Set on the resulting graph. Now observe that if the original graph has bounded mim-width then the resulting graph still has bounded mim-width, since we only added a single vertex. Therefore, all of our algorithms given in Theorems 24 and 25 have the same running times for the Node Multiway Cut problem.

References

[1] Rémy Belmonte and Martin Vatshelle. Graph classes with structured neighborhoods and algorithmic applications. *Theoret. Comput. Sci.*, 511:54–65, 2013.

[2] Benjamin Bergougnoux, Edouard Bonnet, Nick Brettell, and O-joung Kwon. Personal communication, draft available upon request.

[3] Benjamin Bergougnoux and Mamadou Moustapha Kanté. Rank based approach on graphs with structured neighborhood. *CoRR*, abs/1805.11275, 2018.

[4] Benjamin Bergougnoux and Mamadou Moustapha Kanté. Fast exact algorithms for some connectivity problems parameterized by clique-width. *Theor. Comput. Sci.*, 782:30–53, 2019.

[5] Benjamin Bergougnoux and Mamadou Moustapha Kanté. More applications of the d-neighbor equivalence: Connectivity and acyclicity constraints. In *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany.*, pages 17:1–17:14, 2019.

[6] Hans L. Bodlaender. Treewidth: Characterizations, applications, and computations. In *Graph-Theoretic Concepts in Computer Science, 32nd International Workshop, WG 2006, Bergen, Norway, June 22-24, 2006, Revised Papers*, pages 1–14, 2006.

[7] Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inform. and Comput.*, 243:86–111, 2015.

[8] Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theoret. Comput. Sci.*, 511:66–76, 2013.

[9] G. Calinescu. *Multiway Cut.* Springer, 2008.

[10] J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55:1–13, 2009.

[11] R. H. Chitnis, F. V. Fomin, D. Lokshtanov, P. Misra, M. S. Ramanujan, and S. Saurabh. Faster exact algorithms for some terminal set problems. *Journal of Computer and System Sciences*, 88:195–207, 2017.

[12] D. G. Corneil and J. Fonlupt. The complexity of generalized clique covering. *Discrete Applied Mathematics*, 22(2):109 – 118, 1988.

[13] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[14] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

[15] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3:1–3:11, 2013.

[16] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[17] G. Even, J. Naor, and L. Zosin. An 8-approximation algorithm for the subset feedback vertex set problem. *SIAM J. Comput.*, 30(4):1231–1252, 2000.

[18] F. V. Fomin, S. Gaspers, D. Lokshtanov, and S. Saurabh. Exact algorithms via monotone local search. In *Proceedings of STOC 2016*, pages 764–775, 2016.

[19] F. V. Fomin, P. Heggernes, D. Kratsch, C. Papadopoulos, and Y. Villanger. Enumerating minimal subset feedback vertex sets. *Algorithmica*, 69(1):216–231, 2014.

[20] Fedor V. Fomin, Petr A. Golovach, and Jean-Florent Raymond. On the tractability of optimization problems on h-graphs. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 30:1–30:14, 2018.

[21] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.

[22] P. A. Golovach, P. Heggernes, D. Kratsch, and R. Saei. Subset feedback vertex sets in chordal graphs. *J. Discrete Algorithms*, 26:7–15, 2014.

[23] Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *Int. J. Found. Comput. Sci.*, 11(3):423–443, 2000.

[24] E. C. Hols and S. Kratsch. A randomized polynomial kernel for subset feedback vertex set. *Theory Comput. Syst.*, 62:54–65, 2018.

[25] Lars Jaffke, O-joung Kwon, Torstein J. F. Strømme, and Jan Arne Telle. Mim-width III. graph powers and generalized distance domination problems. *Theoretical Computer Science*, to appear.

[26] Lars Jaffke, O-joung Kwon, and Jan Arne Telle. A unified polynomial-time algorithm for feedback vertex set on graphs of bounded mim-width. In *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, pages 42:1–42:14, 2018.

[27] Lars Jaffke, O-joung Kwon, and Jan Arne Telle. Mim-width II. the feedback vertex set problem. *Algorithmica*, to appear.

[28] K. Kawarabayashi and Y. Kobayashi. Fixed-parameter tractability for the subset feedback set problem and the s-cycle packing problem. *J. Comb. Theory, Ser. B*, 102(4):1020–1034, 2012.

[29] Ki Hang Kim. *Boolean matrix theory and applications*, volume 70. Dekker, 1982.

[30] D. Kratsch, H. Müller, and I. Todinca. Feedback vertex set on AT-free graphs. *Discrete Applied Mathematics*, 156(10):1936–1947, 2008.

[31] Sang-il Oum. Rank-width and vertex-minors. *J. Combin. Theory Ser. B*, 95(1):79–100, 2005.

[32] Sang-Il Oum. Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms*, 5(1):Art. 10, 20, 2009.

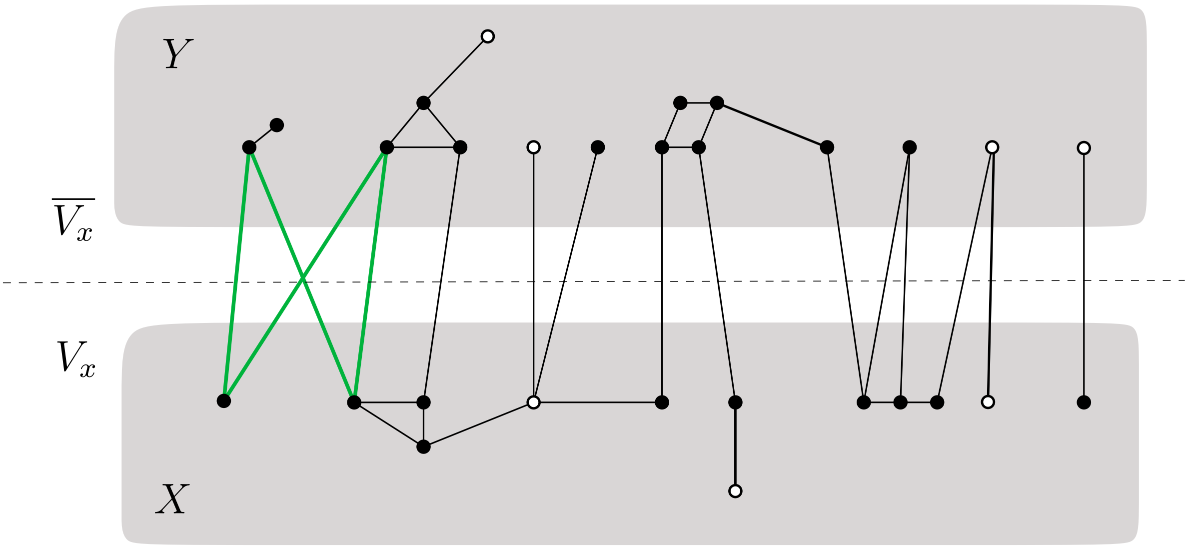
[33] Sang-il Oum, Sigve Hortemo Sæther, and Martin Vatshelle. Faster algorithms for vertex partitioning problems parameterized by clique-width. *Theoret. Comput. Sci.*, 535:16–24, 2014.

[34] Sang-il Oum and Paul Seymour. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B*, 96(4):514–528, 2006.

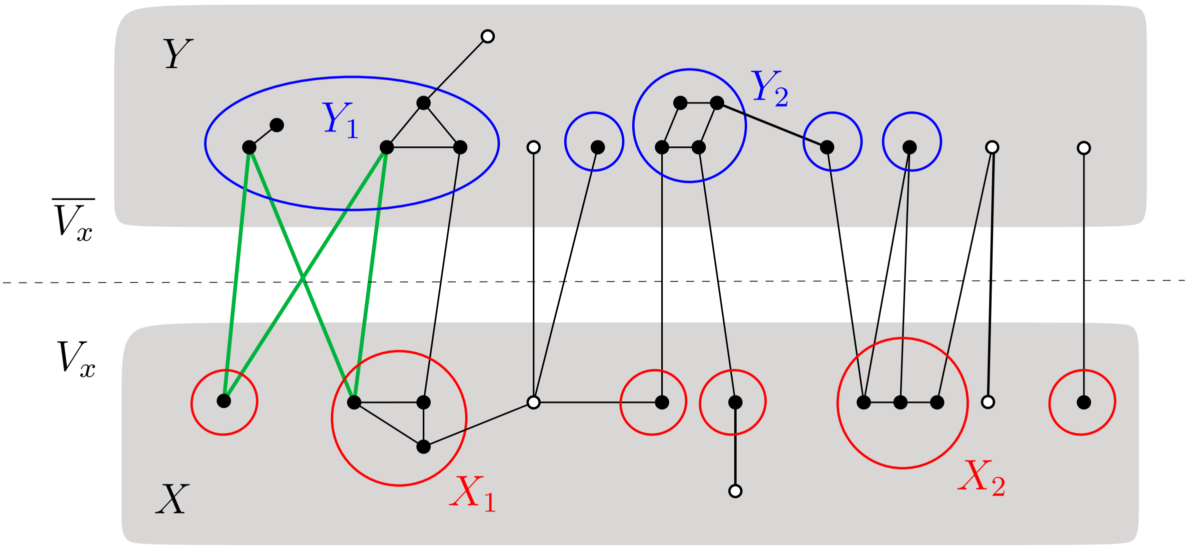
[35] Charis Papadopoulos and Spyridon Tzimas. Subset feedback vertex set on graphs of bounded independent set size. In *13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland*, pages 20:1–20:14, 2018.

[36] Charis Papadopoulos and Spyridon Tzimas. Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs. *Discrete Applied Mathematics*, 258:204–221, 2019.

[37] Martin Vatshelle. *New width parameters of graphs*. PhD thesis, University of Bergen, Bergen, Norway, 2012.

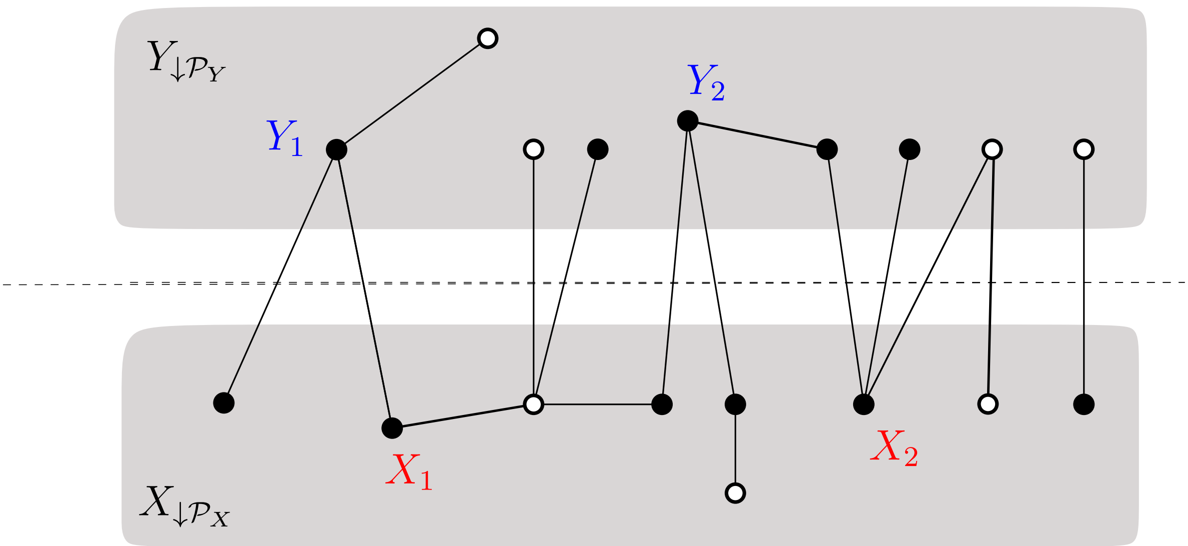


These figures explain the relation between a solution and an index i. We take an S-forest induced by two sets $X \subseteq V_x$ and $Y \subseteq \overline{V_x}$. Vertices in S are those in white.



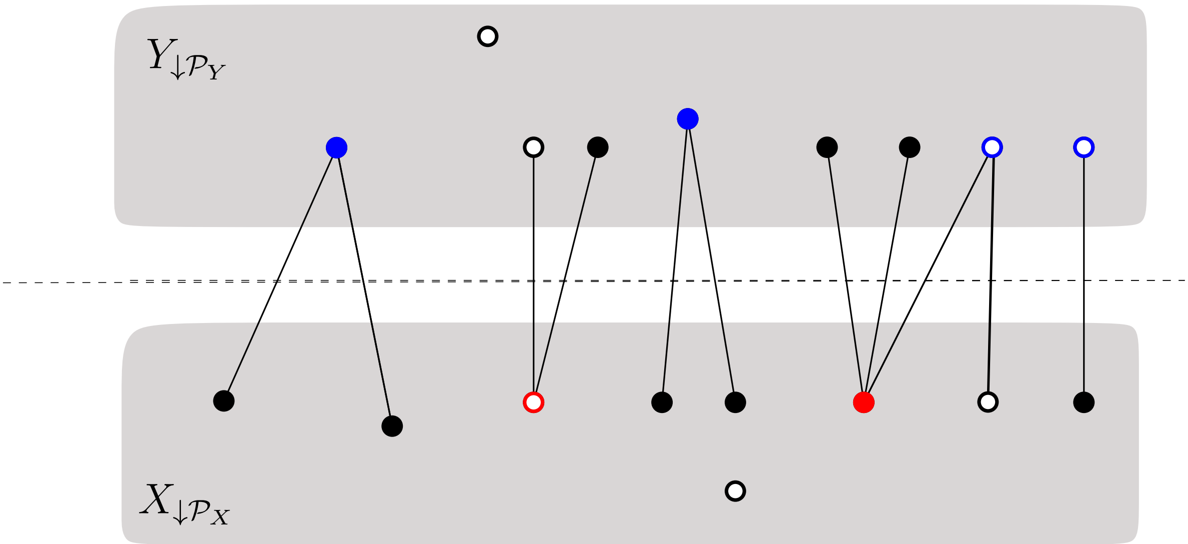
The $\overline{S}$-contractions ($\mathcal{P}_X$ and $\mathcal{P}_Y$) we use to reduce the S-forest into a forest.

Only Y_1 is a non-trivial block: a block which is not a connected component of $G[X \setminus S]$ nor $G[Y \setminus S]$. We need Y_1 to kill the green cycle.

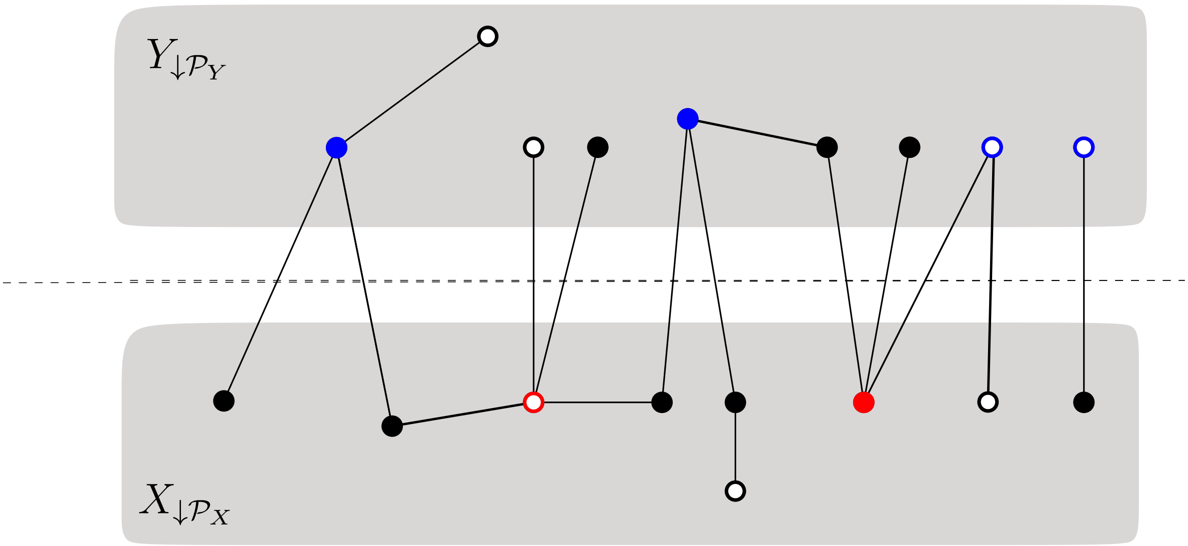


The contracted graph $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$.

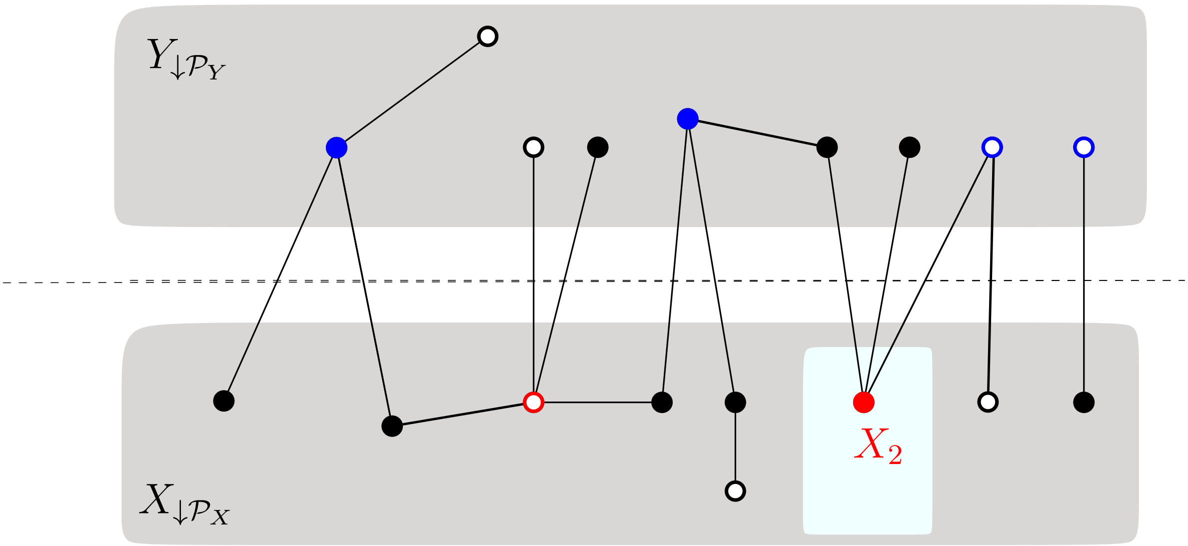
The vertices of this graph are the blocks of $\mathcal{P}_X \cup \mathcal{P}_Y$ and the singletons $\{x\}$ for every vertex x in $(X \cup Y) \cap S$.



The bipartite graph $G[X, Y]_{\downarrow \mathcal{P}_Y \cup \mathcal{P}_X}$. The colored vertices form a vertex cover VC of this graph that satisfies the properties of Lemma 6.



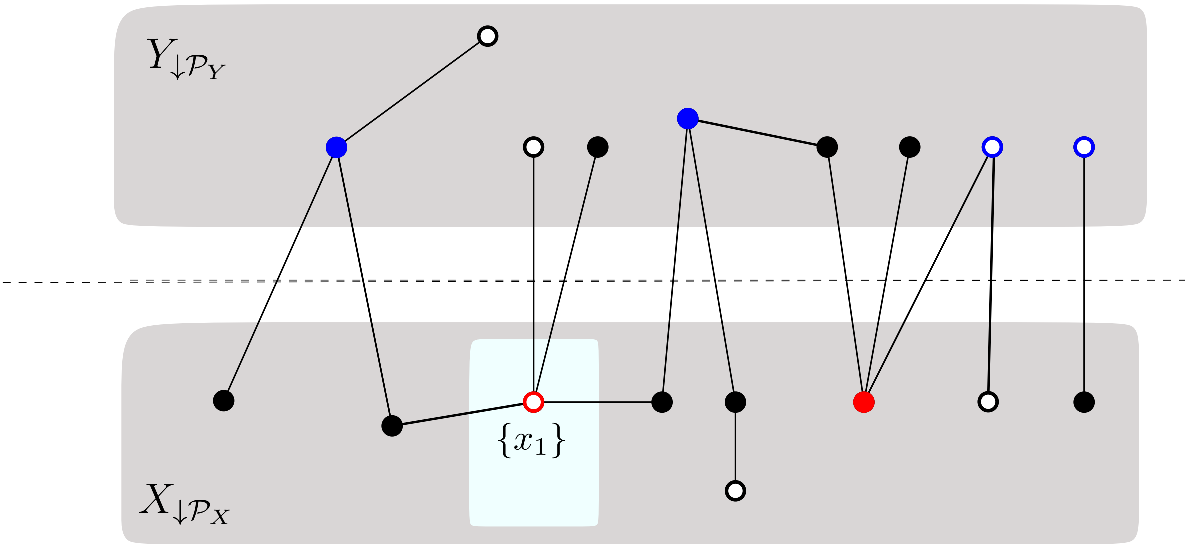
We construct an index $i = (X^{\overline{S}}_{\mathsf{vc}}, X^{S}_{\mathsf{vc}}, X_{\overline{\mathsf{vc}}}, Y^{\overline{S}}_{\mathsf{vc}}, Y^{S}_{\mathsf{vc}}) \in \mathbb{I}_x$ such that X is a partial solution associated with i and Y is a complement solution associated with i.



$X^{\overline{S}}_{\textsf{vc}}$

$X^{\overline{S}}_{\textsf{vc}}$ contains the representatives of the blocks in $\mathcal{P}_X$ that are in the vertex cover VC.

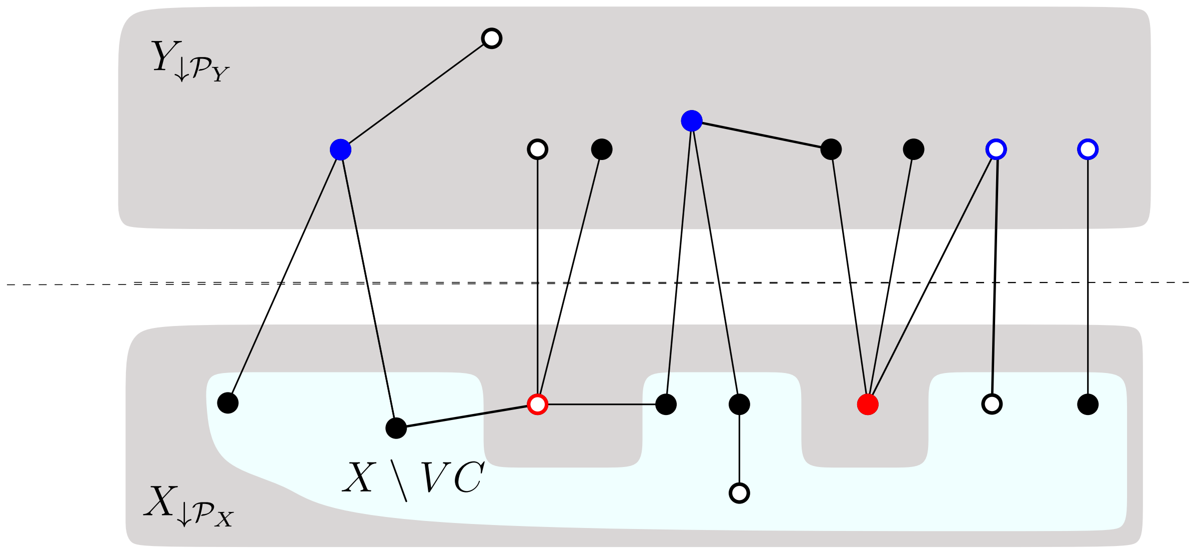
Here $X^{\overline{S}}_{\textsf{vc}} = \{R_2\}$ where $R_2 = \mathsf{rep}^2_{V_x}(X_2)$.



X^S_{vc}

X^S_{vc} contains the representatives of the sigletons $\{x\}$ that belong to VC and such that $x \in S$.

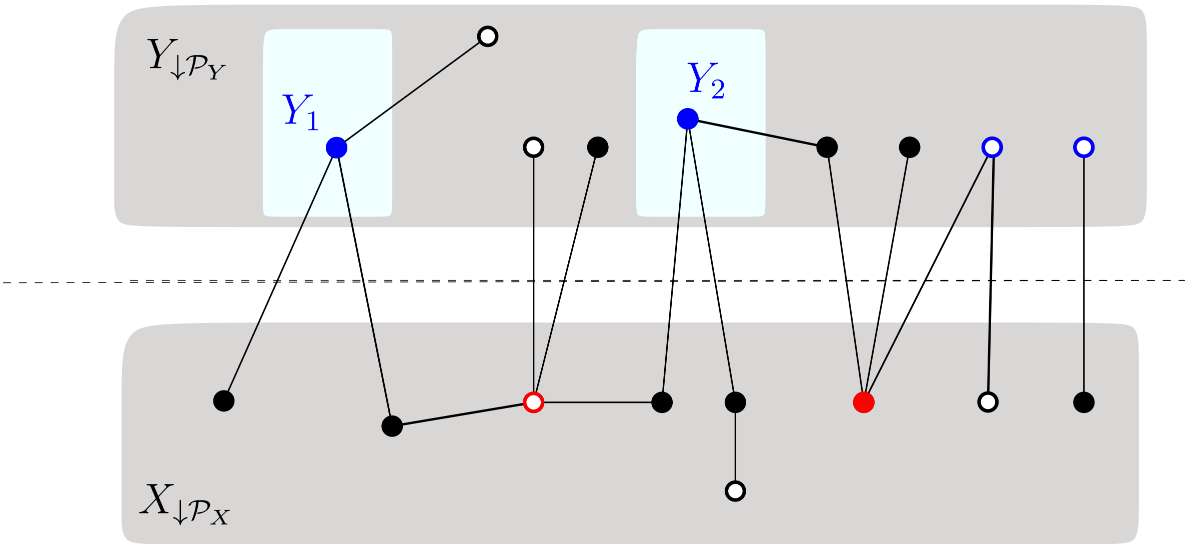
Here, $X^S_{\mathsf{vc}} = \{R_1\}$ where $R_1 = \mathsf{rep}^1_{V_x}(\{x_1\})$.



$X_{\overline{\mathsf{vc}}}$

$X_{\overline{\mathsf{vc}}} = \mathsf{rep}^1_{V_x}(X \setminus VC)$.

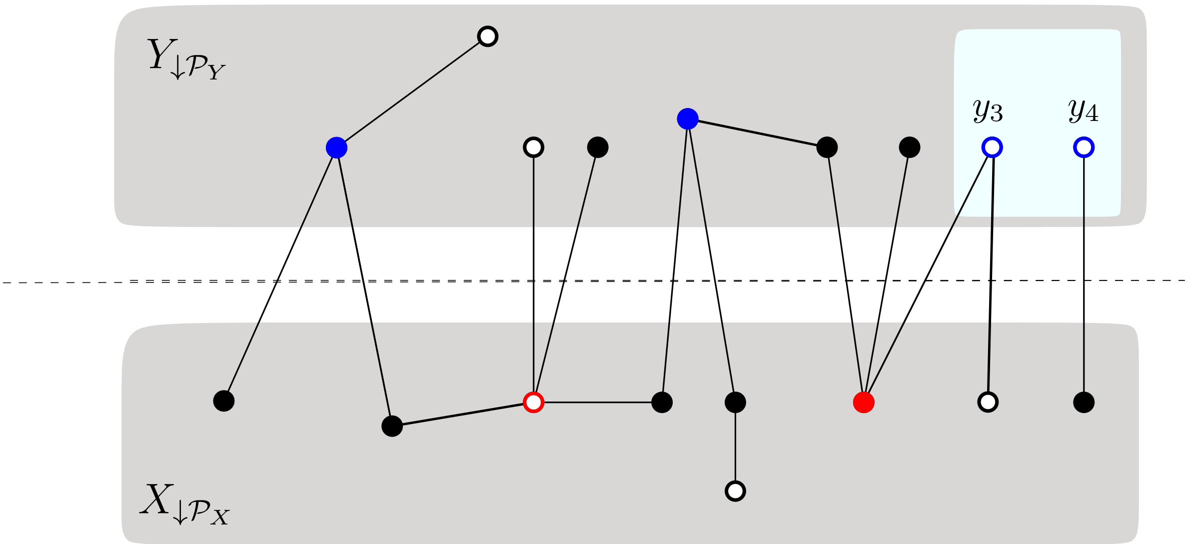
It is the representative set of the vertices of X that are not in a block of VC.



$Y_{\mathsf{vc}}^{\overline{S}}$

$Y_{\mathsf{vc}}^{\overline{S}}$ contains the representatives of the blocks of $\mathcal{P}_Y$ that belong to VC.

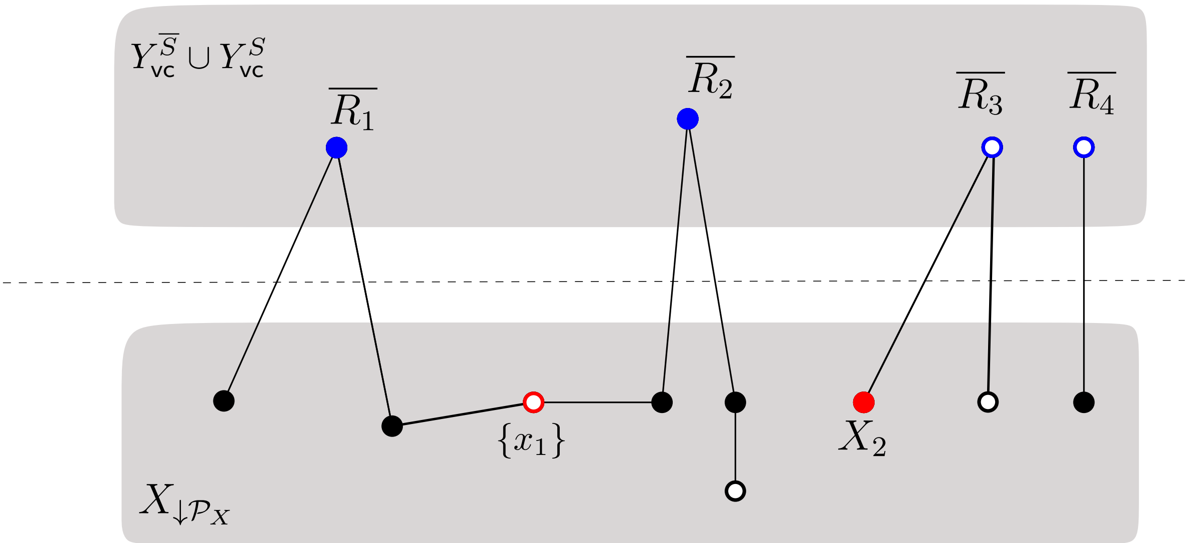
Here $Y_{\mathsf{vc}}^{\overline{S}} = \{\overline{R_1}, \overline{R_2}\}$ where $\overline{R_\ell} = \mathsf{rep}^2_{\overline{V_x}}(Y_\ell)$ for $\ell = 1, 2$.



Y_{vc}^S

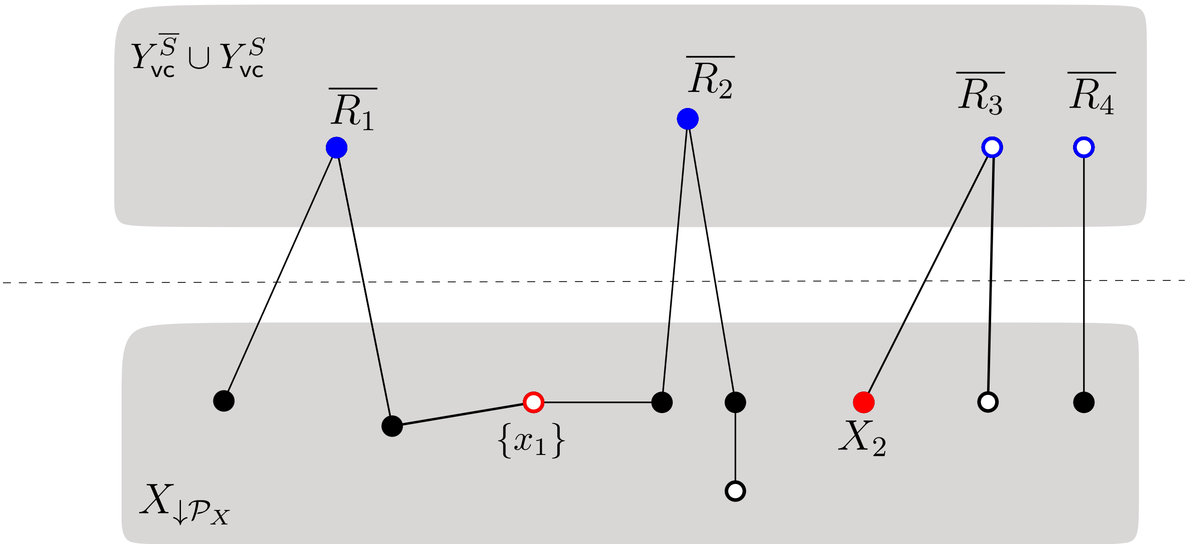
Y_{vc}^S contains the representatives the singletons $\{y\}$ that belong to VC with $y \in S$.

Here $Y_{\mathsf{vc}}^S = \{\overline{R_3}, \overline{R_4}\}$ where $\overline{R_\ell} = \mathsf{rep}^2_{\overline{V_x}}(\{y_\ell\})$ for $\ell = 3, 4$.



The auxiliary graph $\mathrm{aux}_x(X, i)$.

It is obtained from $G[X \cup Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ by (1) removing the edges between blocks of $Y_{\downarrow \mathcal{P}_Y}$, (2) removing the blocks of $Y_{\downarrow \mathcal{P}_Y}$ that do not belong to VC and (3) remplacing each remaining blocks of $Y_{\downarrow \mathcal{P}_Y}$ by its representatives.



$CC(X,i)$ is the partition with blocks $\{R_1, \overline{R_1}, \overline{R_2}\}$, $\{R_2, \overline{R_3}\}$ and $\{\overline{R_4}\}$ where R_1 and R_2 are the representatives of $\{x_1\}$ and X_2 respectively.

A partial solution W associated with i is equivalent to X if $CC(X,i) = CC(W,i)$. For every partial solution W (associated with i) equivalent to X, we have $G[W \cup Y]$ is an S-forest. So it is sufficient to keep the biggest.

University of Bergen, Norway
Email address: `benjamin.bergougnoux@uib.no`

University of Ioannina, Greece
Email address: `charis@cs.uoi.gr`

University of Bergen, Norway
Email address: `Jan.Arne.Telle@uib.no`