



# Node Multiway Cut and Subset Feedback Vertex Set on Graphs of Bounded Mim-Width

Benjamin Bergougnoux<sup>1</sup> · Charis Papadopoulos<sup>2</sup> · Jan Arne Telle<sup>1</sup>

Received: 24 September 2020 / Accepted: 13 January 2022 / Published online: 30 January 2022  
© The Author(s) 2022

## 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 allows 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 82(1):118–145, 2020) 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 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.

**Keywords** Subset feedback vertex set · Node multiway cut · Neighbor equivalence · Rank-width · Mim-width

---

An extended abstract appeared in the proceedings of 46th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2020.

---

✉ Benjamin Bergougnoux  
benjamin.bergougnoux@uib.no

<sup>1</sup> Department of Informatics, University of Bergen, Bergen, Norway

<sup>2</sup> Department of Mathematics, University of Ioannina, Ioannina, Greece

## 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 [30] independently showed that SFVS is fixed-parameter tractable (FPT) parameterized by the solution size, while Hols and Kratsch [25] 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 [10, 18] where the current best is an  $O^*(1.864^n)$  algorithm due to Fomin et al. [20], and restricted to special graph classes [2, 23, 37, 38]. Naturally, FVS and SFVS differ in complexity, as exemplified by split graphs where FVS is polynomial-time solvable [11] whereas SFVS remains NP-hard [20]. Moreover, note that the vertex-weighted variation of SFVS behaves differently than 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 [38].

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 [8, 21]. NMC is a well-studied problem in terms of approximation [21], as well as parameterized algorithms [8–10, 14, 16, 20]. 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 and setting  $S = \{v\}$  [20]. 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. This explains why most of the positive results on SFVS on graph families [23, 37, 38] can not be translated to NMC.

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 [4], clique-width [13], rank-width [33], and maximum induced matching width (*a.k.a.* mim-width) [41]. These are of varying strength, with tree-width of modeling power strictly weaker than clique-width, as it is bounded on a proper 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 [24]. In this way, an XP algorithm parameterized by mim-width has the feature of unifying several algorithms on well-known graph classes.

We obtain most of these parameters through the well-known notion of *branch-decomposition* introduced in [39]. 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 [33] whose cut function is the  $GF[2]$ -rank of the adjacency matrix, (ii)  $\mathbb{Q}$ -rank-width [35] a variant of rank-width with interesting algorithmic properties which instead uses the rank over the rational field, and (iii) mim-width [41] whose cut function is the size of a maximum induced matching of the graph crossing the cut. Concerning their computations, for rank-width and  $\mathbb{Q}$ -rank-width, there are  $2^{3k} \cdot n^4$  time algorithms that, given a graph  $G$  as input and  $k \in \mathbb{N}$ , either output a decomposition for  $G$  of width at most  $3k + 1$  or confirms that the width of  $G$  is more than  $k$  [35, 36]. However, 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  $f$ .

Let us mention what is known regarding the complexity of NMC and SFVS parameterized by these width measures. Since these problems can be expressed in  $\text{MSO}_1$ -logic it follows that they are FPT parameterized by tree-width, clique-width, rank-width or  $\mathbb{Q}$ -rank-width [12, 34], however the runtime will contain a tower of 2's with more than 4 levels. Recently, Bergougnoux et al. [2] proposed  $k^{O(k)} \cdot n^3$  time algorithms for these two problems parameterized by treewidth and proved that they cannot be solved in time  $k^{o(k)} \cdot n^{O(1)}$  unless ETH fails. For mim-width, we know that FVS and thus SFVS are both  $W[1]$ -hard when parameterized by the mim-width of a given decomposition [29].

Attacking SFVS seems to be a hard task that requires more tools than for FVS. Even for very small values of mim-width that capture several graph classes, the tractability of SFVS, prior to our result, was left open besides interval and permutation graphs [37]. Although FVS was known to be tractable on such graphs for more than a decade [32], the complexity status of SFVS still remained unknown.

**Our Results** We design a meta-algorithm that, given a graph and a branch-decomposition, solves SFVS (or NMC via its reduction to SFVS). The runtime of this algorithm is upper bounded by  $2^{O(rw^3)} \cdot n^4$ ,  $2^{O(q^2 \log(q))} \cdot n^4$  and  $n^{O(k^2)}$  where  $rw$ ,  $q$  and  $k$  are the rank-width, the  $\mathbb{Q}$ -rank-width and the mim-width of the given branch-decomposition. For clique-width, our meta-algorithm implies that we can solve SFVS and NMC in time  $2^{O(k^2)} \cdot n^{O(1)}$  where  $k$  is the clique-width of a given clique-width expression. However, we do not prove this as it is not asymptotically optimal, indeed Jacob et al. [26] show recently that SFVS and NMC is solvable in time  $2^{O(k \log k)} \cdot n$  given a clique-width expression.

We resolve in the affirmative the question raised by Jaffke et al. [29], also mentioned in [37, 38], 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  $\text{MSO}_1$  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 [27] 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 [37]. 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 if an intersection model is given as input (from which we can compute a decomposition of mim-width 1) [1, 27] 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  $2|E(H)|$ ) [19].

**Our Approach** We give some intuition to our meta-algorithm, that will focus on SUBSET FEEDBACK VERTEX SET. 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 the  $d$ -neighbor equivalence, with  $d = 1$  and  $d = 2$ , a notion introduced by Bui-Xuan et al. [7]. Two subsets  $X$  and  $Y$  of  $A \subseteq V(G)$  are  $d$ -neighbor equivalent w.r.t.  $A$ , if  $\min(d, |X \cap N(u)|) = \min(d, |Y \cap N(u)|)$  for all  $u \in V(G) \setminus A$ . For a cut  $(A, \bar{A})$  this equivalence relation on subsets of vertices was used by Bui-Xuan et al. [7] to design a meta-algorithm, also giving XP algorithms by mim-width, for so-called  $(\sigma, \rho)$  generalized domination problems. Recently, Bergougnoux and Kanté [3] extended the uses of this notion to acyclic and connected variants of  $(\sigma, \rho)$  generalized domination and similar problems like FVS. An earlier XP algorithm for FVS parameterized by mim-width had been given by Jaffke et al. [29] but instead of the  $d$ -neighbor equivalences this algorithm was based on the notions of 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  $(A, \bar{A})$  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  $Y \subseteq \bar{A}$ , we have  $\text{best}(\mathcal{A}, Y) = \text{best}(\mathcal{B}, Y)$  where  $\text{best}(\mathcal{C}, Y)$  is the maximum weight of a set  $X \in \mathcal{C}$  such that  $X \cup Y$  induces an  $S$ -forest. Since the root of the decomposition is associated with the cut  $(V(G), \emptyset)$ , the set of partial solutions computed for this cut represents  $2^{V(G)}$  and thus contains an  $S$ -forest of maximum weight. 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 [3] or [29]. 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  $\bar{A} \cap V(F)$  could be a biclique. Instead, we introduce a notion of vertex contractions and prove that, for every  $X \subseteq A$  and  $Y \subseteq \bar{A}$ , 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 [3, 29]. 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 us 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 subset of vertices in VC. Moreover, for each index  $i$ , we introduce the notions of partial solutions and complement solutions associated with  $i$  which correspond, respectively, to subsets of  $X \subseteq A$  and subsets  $Y \subseteq \bar{A}$  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  $S$ -forest  $F$ , there exists an index  $i$  such that  $V(F) \cap A$  is a partial solution associated with  $i$  and  $V(F) \cap \bar{A}$  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 a dynamic programming algorithm parameterized by graph width measures uses this notion of vertex contractions. Note that in contrast to the meta-algorithms in [3, 7], 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 [3, 7], i.e.  $n^{O(mim^2)}$  versus  $n^{O(mim)}$ . However, for the case  $S = V(G)$ , thus solving FVS, our meta-algorithm will have runtime  $n^{O(mim)}$ , as the algorithms for FVS of [3, 29]. We do not expect that SFVS can be solved as fast as FVS when parameterized by graph width measures. In fact, we know that it is not the case for tree-width as FVS can be solved in  $2^{O(k)} \cdot n$  [5] but 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  $\min(\emptyset) = +\infty$  and  $\max(\emptyset) = -\infty$ .

**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  $N_G(U) = (\bigcup_{v \in U} N_G(v)) \setminus U$ .

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  $VC \subseteq V(G)$  such that, for every edge  $uv \in E(G)$ , we have  $u \in VC$  or  $v \in VC$ . 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  $cc_G(X)$  the partition  $\{C \subseteq V(G) \mid G[C] \text{ is a connected component of } G[X]\}$ . We will omit the subscript  $G$  of the neighborhood and components notations whenever there is no ambiguity.

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 following equivalent formulation:

**SUBSET FEEDBACK VERTEX SET (SFVS)**

**Input:** A graph  $G$ ,  $S \subseteq V(G)$  and a weight function  $w : V(G) \rightarrow \mathbb{Q}$ .

**Output:** The maximum among the weights of the  $S$ -forests of  $G$ .

**Rooted Layout** For the notion of branch-decomposition, we consider its rooted variant called *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  $(T, \delta)$  of a rooted binary tree  $T$  and a bijective function  $\delta$  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  $V_x$  the set of vertices that are in bijection with  $L_x$ , i.e.,  $V_x := \{v \in V(G) \mid \delta(v) \in L_x\}$ .

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  $f : 2^{V(G)} \rightarrow \mathbb{N}$  and a rooted layout  $\mathcal{L} = (T, \delta)$ , the  $f$ -width of a node  $x$  of  $T$  is  $f(V_x)$  and the  $f$ -width of  $(T, \delta)$ , denoted by  $f(T, \delta)$  (or  $f(\mathcal{L})$ ), is  $\max\{f(V_x) \mid x \in V(T)\}$ . 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  $rw_{\mathbb{Q}}$ -width where  $rw(A)$  (resp.  $rw_{\mathbb{Q}}(A)$ ) is the rank over  $GF(2)$  (resp.  $\mathbb{Q}$ ) of the matrix  $M_{A, \bar{A}}$  for all  $A \subseteq V(G)$ .

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

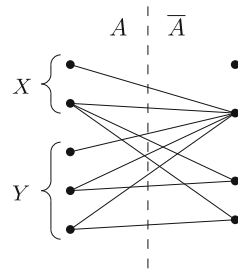
Observe that all three parameters  $rw$ -,  $rw_{\mathbb{Q}}$ -, and  $mim$ -width are symmetric, i.e., for the associated set function  $f$  and for any  $A \subseteq V(G)$ , we have  $f(A) = f(\bar{A})$ . The following lemma provides upper bounds between  $mim$ -width and the other two parameters.

**Lemma 3** ([41]) *Let  $G$  be a graph. For every  $A \subseteq V(G)$ , we have  $mim(A) \leq rw(A)$  and  $mim(A) \leq rw_{\mathbb{Q}}(A)$ .*

**Proof** Let  $A \subseteq V(G)$ . Let  $S$  be the vertex set of a maximum induced matching of the graph  $G[A, \bar{A}]$ . By definition, we have  $mim(A) = |S \cap A| = |S \cap \bar{A}|$ . Observe that the restriction of the matrix  $M_{A, \bar{A}}$  to rows in  $S \cap A$  and columns in  $S \cap \bar{A}$  is a permutation matrix: a binary square matrix with exactly one entry of 1 in each row and each column. The rank of this permutation matrix over  $GF[2]$  or  $\mathbb{Q}$  is  $|S \cap A| = mim(A)$ . Hence,  $mim(A)$  is upper bounded both by  $rw(A)$  and  $rw_{\mathbb{Q}}(A)$ . □

**$d$ -Neighbor-Equivalence** The following concepts were introduced in [7]. 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  $X \equiv_d^A Y$ , if  $\min(d, |X \cap N(u)|) = \min(d, |Y \cap N(u)|)$  for all  $u \in \bar{A}$ . It is not hard to check that  $\equiv_d^A$  is an equivalence relation. See Fig. 1 for an example of 2-neighbor equivalent sets.

**Fig. 1** We have  $X \equiv_2^A Y$ , but it is not the case that  $X \equiv_3^A Y$



For all  $d \in \mathbb{N}^+$ , we let  $\text{necc}_d : 2^{V(G)} \rightarrow \mathbb{N}$  where for all  $A \subseteq V(G)$ ,  $\text{necc}_d(A)$  is the number of equivalence classes of  $\equiv_d^A$ . Notice that  $\text{necc}_1$  is a symmetric function [31, Theorem 1.2.3] but  $\text{necc}_d$  is not necessarily symmetric for  $d \geq 2$ . To simplify the running times, we will use the shorthand  $s\text{-necc}_2(A)$  to denote  $\max(\text{necc}_2(A), \text{necc}_2(\bar{A}))$  (where  $s$  stands for symmetric). The following lemma shows how  $\text{necc}_d(A)$  is upper bounded by the other parameters.

**Lemma 4** ([1, 35, 41]) *Let  $G$  be a graph. For every  $A \subseteq V(G)$  and  $d \in \mathbb{N}^+$ , we have the following upper bounds on  $\text{necc}_d(A)$ :*

- (a)  $2^{\text{drw}(A)^2}$ ,
- (b)  $2^{\text{rw}_Q(A) \log(\text{drw}_Q(A)+1)}$ ,
- (c)  $|A|^{d \text{mim}(A)}$ .

In order to manipulate the equivalence classes of  $\equiv_d^A$ , 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  $\text{rep}_d^A(X)$  the lexicographically smallest set  $R \subseteq A$  such that  $|R|$  is minimized and  $R \equiv_d^A X$ . Moreover, we denote by  $\mathcal{R}_d^A$  the set  $\{\text{rep}_d^A(X) \mid X \subseteq A\}$ . It is worth noticing that the empty set always belongs to  $\mathcal{R}_d^A$ , for all  $A \subseteq V(G)$  and  $d \in \mathbb{N}^+$ . Moreover, we have  $\mathcal{R}_d^{V(G)} = \mathcal{R}_d^\emptyset = \{\emptyset\}$  for all  $d \in \mathbb{N}^+$ . In order to compute these representatives, we use the following lemma.

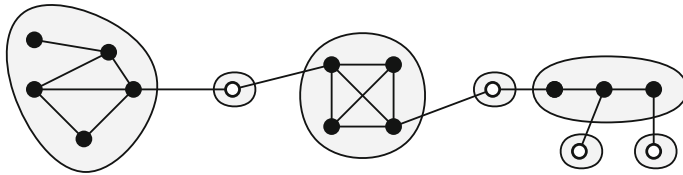
**Lemma 5** ([7]) *Let  $G$  be an  $n$ -vertex graph. For every  $A \subseteq V(G)$  and  $d \in \mathbb{N}^+$ , one can compute in time  $O(\text{necc}_d(A) \cdot n^2 \cdot \log(\text{necc}_d(A)))$ , the sets  $\mathcal{R}_d^A$  and a data structure that, given a set  $X \subseteq A$ , computes  $\text{rep}_d^A(X)$  in time  $O(|A| \cdot n \cdot \log(\text{necc}_d(A)))$ .*

**Vertex Contractions** In order to deal with SFVS, we will use the ideas of the algorithms for FEEDBACK VERTEX SET from [3, 28]. To this end, we will contract subsets of  $\bar{S}$  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  $\mathcal{R}_2^A$ . Since the sets in  $\mathcal{R}_2^A$  are not necessarily pairwise disjoint, we will use the following notions of graphs “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  $N(A) \cap B \neq \emptyset$ . Observe that if the





**Fig. 2** An  $S$ -forest induced by a set  $X \subseteq V(G)$ , the vertices of  $S$  are white. The gray circles represent the blocks of  $X_{\downarrow cc(X \setminus S)}$ . The graph  $G[X]_{\downarrow cc(X \setminus S)}$  is obtained by contracting each gray circle

sets in  $\mathcal{A}$  are pairwise disjoint, then  $G[\mathcal{A}]$  is obtained from an induced subgraph of  $G$  by *vertex contractions* (i.e., by replacing two vertices  $u$  and  $v$  with a new vertex with neighborhood  $N(\{u, v\})$ ) and, for this reason, we refer to  $G[\mathcal{A}]$  as a *contracted graph*. Notice that we will never use the neighborhood notation and connected component notations on contracted graphs. 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  $N(A) \cap B \neq \emptyset$ . 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 observation 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.

**Observation 6** *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  $\bar{S}$ -contraction of  $X$ . We will use the following notations to handle these contractions.

Given  $Y \subseteq V(G)$ , we denote by  $\binom{Y}{1}$  the partition of  $Y$  which contains only singletons, i.e.,  $\binom{Y}{1} = \{\{v\} \mid v \in Y\}$ . Moreover, for an  $\bar{S}$ -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  $H$  of  $G$  and an  $\bar{S}$ -contraction  $\mathcal{P}$  of  $V(H)$ , we denote by  $H_{\downarrow \mathcal{P}}$  the graph  $H[V(H)_{\downarrow \mathcal{P}}]$ . For example, given two  $\bar{S}$ -contractions  $\mathcal{P}_X, \mathcal{P}_Y$  of two disjoint subsets  $X, Y$  of  $V(G)$ , we denote the graph  $G[X_{\downarrow \mathcal{P}_X}, Y_{\downarrow \mathcal{P}_Y}]$  by  $G[X, Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$  and the graph  $G[X_{\downarrow \mathcal{P}_X} | Y_{\downarrow \mathcal{P}_Y}]$  by  $G[X | Y]_{\downarrow \mathcal{P}_X \cup \mathcal{P}_Y}$ . It is worth noticing that in our contracted graphs, all the blocks of  $S$ -vertices are singletons and we denote them by  $\{v\}$ .

Given a set  $X \subseteq V(G)$ , we will intensively use the graph  $G[X]_{\downarrow cc(X \setminus S)}$  which corresponds to the graph obtained from  $G[X]$  by contracting the connected components of  $G[X \setminus S]$ , see Fig. 2.

Observe that, for every subset  $X \subseteq V(G)$ , if  $G[X]$  is an  $S$ -forest, then  $G[X]_{\downarrow cc(X \setminus S)}$  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.

**Fact 7** *Let  $G$  be a graph and  $S \subseteq V(G)$ . For every  $X \subseteq V(G)$ ,  $G[X]$  is an  $S$ -forest if and only if there exists an  $\bar{S}$ -contraction  $\mathcal{P}$  of  $X$  satisfying the following two properties:*

- $G[X]_{\downarrow \mathcal{P}}$  is a forest, and
- for every  $B \in \mathcal{P}$  and  $v \in X \cap S$ , we have  $|N(v) \cap B| \leq 1$ .

Moreover, if  $G[X]$  is an  $S$ -forest, then the  $\bar{S}$ -contraction  $\text{cc}(X \setminus S)$  satisfies these two properties.

**Proof** ( $\Rightarrow$ ) Suppose first that  $G[X]$  is an  $S$ -forest. We claim that the  $\bar{S}$ -contraction  $\text{cc}(X \setminus S)$  satisfies the two properties. Assume towards a contradiction that there is a cycle  $C$  in  $G[X]_{\downarrow \text{cc}(X \setminus S)}$ . By definition of  $G[X]_{\downarrow \text{cc}(X \setminus S)}$ , the blocks of this graph are the connected components  $\text{cc}(X \setminus S)$  and the singletons in  $\binom{X \cap S}{1}$ . The blocks in  $\text{cc}(X \setminus S)$  are pairwise non-adjacent, thus  $C$  contains a block  $\{s\}$  with  $s \in X \cap S$ . Observe that for every pair of consecutive blocks  $B_1, B_2$  of  $C$ , there exists a vertex  $v_1 \in B_1$  and  $v_2 \in B_2$  such that  $v_1 v_2 \in E(G[X])$ . As every block of  $G[X]_{\downarrow \text{cc}(X \setminus S)}$  induces a connected component in  $G$ , we can construct an  $S$ -cycle in  $G[X]$  by replacing every block of  $C$  by a path in  $G[X]$ , yielding a contradiction. Hence,  $G[X]_{\downarrow \text{cc}(X \setminus S)}$  is a forest, i.e. the first property is satisfied. Observe that if there exists  $C \in \text{cc}(X \setminus S)$  and  $v \in X \cap S$  such that  $v$  has two neighbors in  $C$ , then there exists an  $S$ -cycle in  $G[X]$  since  $C$  is a connected component. Hence,  $\text{cc}(X \setminus S)$  satisfies the second property.

( $\Leftarrow$ ) Let  $\mathcal{P}$  be a  $\bar{S}$  contraction of a subset  $X \subseteq V(G)$  that satisfies the two properties. Assume for contradiction that there is an  $S$ -cycle  $C$  in  $G[X]$ . Let  $v$  be a vertex of  $C$  that belongs to  $S$  and let  $u$  and  $w$  be the neighbors of  $v$  in  $C$ .

Let  $B_u$  and  $B_w$  be the blocks in  $(X \cup Y)_{\downarrow \mathcal{P}}$  that contain  $u$  and  $w$  respectively. As  $v$  is in  $S$ , it belongs to the block  $\{v\}$  of  $G[X \cup Y]_{\downarrow \mathcal{P}}$  and thus it is not contained in  $B_u$  nor  $B_w$ . The second property implies that  $|N(v) \cap B| \leq 1$  for each  $B \in \mathcal{P}$ . Thus,  $B_u$  and  $B_w$  are two distinct blocks both connected to the block  $\{v\}$ . Since there exists a path between  $u$  and  $w$  in  $C$  that does not go through  $v$ , we deduce that there is a path between  $B_u$  and  $B_w$  in  $G[X]_{\downarrow \mathcal{P}}$  that does not go through  $\{v\}$ . Indeed, this follows from the fact that if there is an edge between two vertices  $a$  and  $b$  in  $G[X]$ , then either  $a$  and  $b$  belong to the same block of  $G[X]_{\downarrow \mathcal{P}}$  or there exists an edge between the blocks in  $G[X]_{\downarrow \mathcal{P}}$  which contain  $a$  and  $b$ . We conclude that there exists a cycle in  $G[X]_{\downarrow \mathcal{P}}$ , a contradiction with the first property.  $\square$

### 3 A Meta-algorithm for Subset Feedback Vertex Set

In the following, we present a meta-algorithm that, given a rooted layout  $(T, \delta)$  of  $G$ , solves SFVS. We will show that this meta-algorithm will imply that SFVS can be solved in time  $2^{O(\text{rw}_{\mathbb{Q}}(G)^2 \log(\text{rw}_{\mathbb{Q}}(G)))} \cdot n^4$ ,  $2^{O(\text{rw}(G)^3)} \cdot n^4$  and  $n^{O(\text{mim}(T, \delta)^2)}$ . The main idea of this algorithm is to use  $\bar{S}$ -contractions in order to employ similar properties of the algorithm for MAXIMUM INDUCED TREE of [3] and the  $n^{O(\text{mim}(T, \delta))}$  time algorithm for FEEDBACK VERTEX SET of [28]. In particular, we use the following

lemma which is proved implicitly in [3]. To simplify the following statements, we fix a graph  $G$ , a rooted layout  $(T, \delta)$  of  $G$  and a node  $x \in V(T)$ .

**Lemma 8** *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  $\{X_0, X_1\}$  of  $X^{2+}$  such that, for each  $i \in \{0, 1\}$  and, for each  $v \in X_i$ , there exists  $y_v \in Y$  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  $\{X_0, X_1\}$ , then  $|X_0| \leq w$  and  $|X_1| \leq w$ . Indeed, if  $F$  admits a good bipartition  $\{X_0, X_1\}$ , then, for each  $i \in \{0, 1\}$ , 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  $x_0 \in C \cap X^{2+}$ . 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  $x_0$  is even because  $F[C]$  is bipartite w.r.t.  $(C \cap X^{2+}, C \cap Y)$ . Let  $X_0$  (resp.  $X_1$ ) be the set of all vertices  $v \in C \cap X^{2+}$  at distance  $2\ell$  from  $x_0$  with  $\ell$  even (resp. odd). We claim that  $\{X_0, X_1\}$  is a good bipartition of  $F[C]$ .

Let  $i \in \{0, 1\}$ ,  $v \in X_i$  and  $\ell \in \mathbb{N}$  such that the distance between  $v$  and  $x_0$  in  $F[C]$  is  $2\ell$ . 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  $v$  has a neighbor  $y$  in  $F$  that is not adjacent to  $P \cap X_i$ . Observe that, for every  $v' \in P$ , the distance between  $v'$  and  $x_0$  in  $F[C]$  is either  $2\ell - 2$ ,  $2\ell$  or  $2\ell + 2$  because  $F[C]$  is a tree and the distance between  $v$  and  $x_0$  is  $2\ell$ . By construction of  $\{X_0, X_1\}$ , every vertex at distance  $2\ell - 2$  or  $2\ell + 2$  from  $x_0$  belongs to  $X_{1-i}$ . Thus, every vertex in  $P \cap X_i$  is at distance  $2\ell$  from  $x_0$ . If  $\ell = 0$ , then we are done because  $v = x_0$  and  $P \cap X_i = \emptyset$ . Assume that  $\ell \neq 0$ . As  $F[C]$  is a tree,  $v$  has only one neighbor  $w$  at distance  $2\ell - 1$  from  $x_0$  in  $F[C]$ . Because  $F[C]$  is a tree,  $w$  is the only vertex adjacent to  $v$  and the vertices in  $P \cap X_i$ . By definition of  $X^{2+}$ ,  $v$  has at least two neighbors in  $Y$ , so  $v$  admits a neighbor that is not  $w$  and this neighbor is not adjacent to the vertices in  $P \cap X_i$ . Hence, we deduce that  $\{X_0, X_1\}$  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. This proves that  $|X^{2+}| \leq 2w$ . □

The following lemma generalizes Fact 7 and presents the equivalence between  $S$ -forests and forests that we will use in our algorithm.

**Lemma 9** *Let  $X \subseteq V_x$  and  $Y \subseteq \overline{V_x}$ . 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)  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  is a forest,
- (2) for all block  $P \in \text{cc}(X \setminus S) \cup \mathcal{P}_Y$  and  $v \in (X \cup Y) \cap S$ , we have  $|N(v) \cap P| \leq 1$ ,
- (3) the graph  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  admits a vertex cover  $\text{VC}$  of size at most  $4\text{mim}(V_x)$  such that the neighborhoods of the blocks in  $\text{VC}$  are pairwise distinct in  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$ .

**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  $\mathcal{P}_Y = cc(Y \setminus S)$ . Observe that there is no cycle in  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  that contains a block in  $\binom{S}{1}$  because  $G[X \cup Y]$  is an  $S$ -forest. Moreover,  $cc(X \setminus S)$  and  $\mathcal{P}_Y$  form two independent sets in  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ . Consequently, for all the cycles  $C$  in  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  we have  $C = (X_1, Y_1, X_2, Y_2, \dots, X_t, Y_t)$  where  $X_1, \dots, X_t \in cc(X \setminus S)$  and  $Y_1, \dots, Y_t \in \mathcal{P}_Y$ . We do the following operation, until the graph  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  is a forest: take a cycle  $C = (X_1, Y_1, X_2, Y_2, \dots, X_t, Y_t)$  in  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  and replace the blocks  $Y_1, \dots, Y_t$  in  $\mathcal{P}_Y$  by the block  $Y_1 \cup \dots \cup Y_t$ . See Figs. 5 and 6 in “Appendix A” for an example of  $\bar{S}$ -contraction  $\mathcal{P}_Y$ .

For each  $B \in cc(X \setminus S) \cup cc(Y \setminus S)$ , the vertices of  $B$  are pairwise connected in  $G[(X \cup Y) \setminus S]$ . We deduce by induction that whenever we apply the operation on a cycle  $C = (X_1, Y_1, X_2, Y_2, \dots, X_t, Y_t)$ , it holds that the vertices of the new block  $Y_1 \cup \dots \cup Y_t$  are pairwise connected in  $G[(X \cup Y) \setminus S]$ . Thus, for every block  $B$  of  $cc(X \setminus S) \cup \mathcal{P}_Y$ , the vertices of  $B$  are pairwise connected in  $G[(X \cup Y) \setminus S]$ . It follows that for every  $v \in (X \cup Y) \cap S$  and  $B \in cc(X \setminus S) \cup \mathcal{P}_Y$ , since  $G[X \cup Y]$  is an  $S$ -forest, we have  $|N(v) \cap B| \leq 1$ . Thus, Condition (2) is satisfied.

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

- the blocks that have at least 2 neighbors in  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ , and
- one block in every isolated edge of  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ .

By construction, it is clear that  $VC$  is indeed a vertex cover of  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  as every edge is either isolated or incident to a block of degree at least 2. We claim that  $|VC| \leq 4mim(V_x)$ . By Observation 6, we know that the size of a maximum induced matching in  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  is at most  $mim(V_x)$ . Let  $t$  be the number of isolated edges in  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ . Observe that the size of a maximum induced matching in the graph obtained from  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  by removing isolated edges is at most  $mim(V_x) - t$ . By Lemma 8, we know that there are at most  $4(mim(V_x) - t)$  blocks that have at least 2 neighbors in  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ . We conclude that  $|VC| \leq 4mim(V_x)$ .

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

In the following, we will use Lemma 9 to design some sort of equivalence relation between partial solutions. To this purpose, we use the following set of tuples. We call each such tuple an index because it corresponds to an index into a table in a DP (dynamic programming) approach. We do this even though the presentation of our algorithm is not given by a standard DP description.

**Definition 10** ( $\mathbb{I}_x$ ) We define the set  $\mathbb{I}_x$  of indices as the set of tuples

$$(x_{vc}^{\bar{S}}, x_{vc}^S, x_{vc}^{\overline{\bar{S}}}, y_{vc}^{\bar{S}}, y_{vc}^S) \in 2^{\mathcal{R}_2^{V_x}} \times 2^{\mathcal{R}_1^{V_x}} \times \mathcal{R}_1^{V_x} \times 2^{\mathcal{R}_2^{\overline{V_x}}} \times 2^{\mathcal{R}_1^{\overline{V_x}}}$$

such that  $|x_{vc}^{\bar{S}}| + |x_{vc}^S| + |y_{vc}^{\bar{S}}| + |y_{vc}^S| \leq 4mim(V_x)$ .

These sets of indices play a major role in our meta-algorithm, in particular, the sizes of these sets of indices appear in the runtime of our meta-algorithm. In fact, to prove the algorithmic consequences of our meta-algorithm for rank-width,  $\mathbb{Q}$ -rank-width and mim-width, we show (Lemma 23) that the size of  $\mathbb{I}_x$  is upper bounded by  $2^{O(\text{rw}(V_x)^3)}$ ,  $2^{O(\text{rw}_{\mathbb{Q}}(V_x)^2 \log(\text{rw}_{\mathbb{Q}}(V_x)))}$  and  $n^{O(\text{mim}(V_x)^2)}$ .

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 itself 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  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  is a forest, then  $G[X \cup Y]$  is an  $S$ -forest.

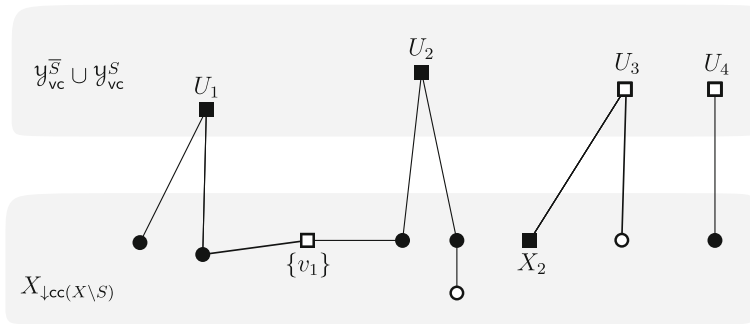
Let us give some intuition on these indices by explaining how one index is associated with a solution, figures explaining this association and the purposes of the sets  $\mathcal{X}_{\text{vc}}^{\overline{S}}, \mathcal{X}_{\text{vc}}^S, \mathcal{X}_{\overline{\text{vc}}}, \mathcal{Y}_{\text{vc}}^{\overline{S}}, \mathcal{Y}_{\text{vc}}^S$  can be found in “Appendix A”. Let  $X \subseteq V_x$  and  $Y \subseteq \overline{V}_x$  such that  $G[X \cup Y]$  is an  $S$ -forest. Let  $\mathcal{P}_Y$  be the  $\overline{S}$ -contraction of  $Y$  and  $\text{VC}$  be a vertex cover of  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  given by Lemma 9. Then,  $X$  and  $Y$  are associated with  $i = (\mathcal{X}_{\text{vc}}^{\overline{S}}, \mathcal{X}_{\text{vc}}^S, \mathcal{X}_{\overline{\text{vc}}}, \mathcal{Y}_{\text{vc}}^{\overline{S}}, \mathcal{Y}_{\text{vc}}^S) \in \mathbb{I}_x$  such that:

- $\mathcal{X}_{\text{vc}}^S$  (resp.  $\mathcal{Y}_{\text{vc}}^S$ ) contains the representatives of the blocks  $\{v\}$  in  $\text{VC}$  such that  $v \in X \cap S$  (resp.  $v \in Y \cap S$ ) w.r.t. the 1-neighbor equivalence over  $V_x$  (resp.  $\overline{V}_x$ ). We will only use the indices where  $\mathcal{X}_{\text{vc}}^S$  contains representatives of singletons, in other words,  $\mathcal{X}_{\text{vc}}^S$  is included in  $\{\text{rep}_1^{V_x}(\{v\}) \mid v \in V_x\}$  which can be much smaller than  $\mathcal{R}_1^{V_x}$ . The same observation holds for  $\mathcal{Y}_{\text{vc}}^S$ . In Definition 10, we state that  $\mathcal{X}_{\text{vc}}^S$  and  $\mathcal{Y}_{\text{vc}}^S$  are, respectively, subsets of  $2^{\mathcal{R}_1^{V_x}}$  and  $2^{\mathcal{R}_1^{\overline{V}_x}}$ , for the sake of simplicity.
- $\mathcal{X}_{\text{vc}}^{\overline{S}}$  (resp.  $\mathcal{Y}_{\text{vc}}^{\overline{S}}$ ) contains the representatives of the blocks in  $\text{cc}(X \setminus S) \cap \text{VC}$  (resp.  $\mathcal{P}_Y \cap \text{VC}$ ) w.r.t. the 2-neighbor equivalence relation over  $V_x$  (resp.  $\overline{V}_x$ ).
- $\mathcal{X}_{\overline{\text{vc}}}$  is the representative of  $X \setminus V(\text{VC})$  (the set of vertices which do not belong to the vertex cover) w.r.t. the 1-neighbor equivalence over  $V_x$ .

Because the neighborhoods of the blocks in  $\text{VC}$  are pairwise distinct in  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  (Property (3) of Lemma 9), there is a one to one correspondence between the representatives in  $\mathcal{X}_{\text{vc}}^{\overline{S}} \cup \mathcal{X}_{\text{vc}}^S \cup \mathcal{Y}_{\text{vc}}^{\overline{S}} \cup \mathcal{Y}_{\text{vc}}^S$  and the blocks in  $\text{VC}$ .

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

Observe that the sets  $\mathcal{X}_{\text{vc}}^{\overline{S}}$  and  $\mathcal{Y}_{\text{vc}}^{\overline{S}}$  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  $\overline{R} \in \mathcal{Y}_{\text{vc}}^{\overline{S}}$ . Thanks to the 2-neighbor equivalence, a vertex  $v$  in  $X \cap S$  has at most one neighbor in  $\overline{R} \in \mathcal{Y}_{\text{vc}}^{\overline{S}}$



**Fig. 3** An example of a set  $X \subseteq V_x$  and its auxiliary graph associated with the index  $i = (\mathcal{X}_{vc}^{\bar{S}}, \mathcal{X}_{vc}^S, \mathcal{X}_{\bar{vc}}, \mathcal{Y}_{vc}^{\bar{S}}, \mathcal{Y}_{vc}^S)$ . Here,  $\mathcal{X}_{vc}^{\bar{S}} = \{R_2\}$  with  $R_2 = \text{rep}_2^{V_x}(X_2)$ ,  $\mathcal{X}_{vc}^S = \{R_1\}$  with  $R_1 = \text{rep}_1^{V_x}(\{v_1\})$ ,  $\mathcal{X}_{\bar{vc}}$  is the representative of the union of the circular blocks,  $\mathcal{Y}_{vc}^{\bar{S}} = \{U_1, U_2\}$ , and  $\mathcal{Y}_{vc}^S = \{U_3, U_4\}$ . The singletons in  $X \cap S$  and  $\mathcal{Y}_{vc}^S$  are white filled, whereas the square blocks are  $\{v_1\}$ ,  $X_2$  and the blocks in  $\mathcal{Y}_{vc}^{\bar{S}} \cup \mathcal{Y}_{vc}^S$

if and only if  $v$  has at most one neighbor in the block of  $\mathcal{P}_Y$  associated with  $\bar{R}$ . This property of the 2-neighbor equivalence is captured by the following fact.

**Fact 11** For every  $A \subseteq V(G)$  and  $B, P \subseteq A$ , if  $B \equiv_2^A P$ , then, for all  $v \in \bar{A}$ , we have  $|N(v) \cap B| \leq 1$  if and only if  $|N(v) \cap P| \leq 1$ .

In order to define partial solutions associated with  $i$ , we need the following notion of auxiliary graph. Given  $X \subseteq V_x$  and  $i = (\mathcal{X}_{vc}^{\bar{S}}, \mathcal{X}_{vc}^S, \mathcal{X}_{\bar{vc}}, \mathcal{Y}_{vc}^{\bar{S}}, \mathcal{Y}_{vc}^S) \in \mathbb{I}_x$ , we write  $\text{aux}(X, i)$  to denote the graph

$$G[X_{\downarrow \text{cc}(X \setminus S)} \mid \mathcal{Y}_{vc}^{\bar{S}} \cup \mathcal{Y}_{vc}^S].$$

Observe that  $\text{aux}(X, i)$  is obtained from the graph induced by  $X_{\downarrow \text{cc}(X \setminus S)} \cup \mathcal{Y}_{vc}^{\bar{S}} \cup \mathcal{Y}_{vc}^S$  by removing the edges between the blocks from  $\mathcal{Y}_{vc}^{\bar{S}} \cup \mathcal{Y}_{vc}^S$ . Figure 3 illustrates an example of the graph  $\text{aux}(X, i)$  and the related notions. The figures in ‘‘Appendix A’’ explain the relations between an  $S$ -forest and these auxiliary graphs.

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

**Definition 12 (Partial solutions)** Let  $i = (\mathcal{X}_{vc}^{\bar{S}}, \mathcal{X}_{vc}^S, \mathcal{X}_{\bar{vc}}, \mathcal{Y}_{vc}^{\bar{S}}, \mathcal{Y}_{vc}^S) \in \mathbb{I}_x$ . We say that  $X \subseteq V_x$  is a partial solution associated with  $i$  if the following conditions are satisfied:

- (a) for every  $R \in \mathcal{X}_{vc}^S$ , there exists a unique  $v \in X \cap S$  such that  $R \equiv_1^{V_x} \{v\}$ ,
- (b) for every  $R \in \mathcal{X}_{vc}^{\bar{S}}$ , there exists a unique  $C \in \text{cc}(X \setminus S)$  such that  $R \equiv_2^{V_x} C$ ,
- (c)  $\text{aux}(X, i)$  is a forest,
- (d) for every  $C \in \text{cc}(X \setminus S)$  and  $\{v\} \in \mathcal{Y}_{vc}^S$ , we have  $|N(v) \cap C| \leq 1$ ,
- (e) for every  $v \in X \cap S$  and  $U \in \mathcal{Y}_{vc}^{\bar{S}} \cup \text{cc}(X \setminus S)$ , we have  $|N(v) \cap U| \leq 1$ ,

- (f)  $\mathcal{X}_{\overline{vc}} \equiv_1^{V_x} X \setminus V(\mathbf{VC}_X)$ , where  $\mathbf{VC}_X$  contains the blocks  $\{v\} \in \binom{X \cap S}{1}$  such that  $\text{rep}_1^{V_x}(\{v\}) \in \mathcal{X}_{\overline{vc}}^S$  and the components  $C$  of  $G[X \setminus S]$  such that  $\text{rep}_2^{V_x}(C) \in \mathcal{X}_{\overline{vc}}^S$ .

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  $i = (\mathcal{X}_{\overline{vc}}^S, \mathcal{X}_{\overline{vc}}^S, \mathcal{X}_{\overline{vc}}, \mathcal{Y}_{\overline{vc}}^S, \mathcal{Y}_{\overline{vc}}^S) \in \mathbb{I}_x$ . We call complement solutions associated with  $i$  all the pairs  $(Y, \mathcal{P}_Y)$  such that  $Y \subseteq \overline{V_x}$ ,  $\mathcal{P}_Y$  is an  $\overline{S}$ -contraction of  $Y$  and the following conditions are satisfied:

- (a) for every  $U \in \mathcal{Y}_{\overline{vc}}^S$ , there exists a unique  $v \in Y \cap S$  such that  $U \equiv_1^{V_x} \{v\}$ ,
- (b) for every  $U \in \mathcal{Y}_{\overline{vc}}^S$ , 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  $\{v\} \in \mathcal{X}_{\overline{vc}}^S$ , we have  $|N(v) \cap P| \leq 1$ ,
- (e) for every  $y \in Y \cap S$  and  $R \in \mathcal{X}_{\overline{vc}}^S \cup \mathcal{P}_Y$ , we have  $|N(y) \cap R| \leq 1$ ,
- (f)  $N(\mathcal{X}_{\overline{vc}}) \cap V(\mathcal{Y}_{\overline{vc}}) = \emptyset$ , where  $\mathcal{Y}_{\overline{vc}}$  contains the blocks  $\{v\} \in \binom{Y \cap S}{1}$  such that  $\text{rep}_1^{V_x}(\{v\}) \notin \mathcal{Y}_{\overline{vc}}^S$  and the blocks  $P \in \mathcal{P}_Y$  such that  $\text{rep}_2^{V_x}(P) \notin \mathcal{Y}_{\overline{vc}}^S$ .

Let us give some explanations on the conditions of Definitions 12 and 13. Let  $X$  be a partial solution associated with an index  $i \in \mathbb{I}_x$  and  $(Y, \mathcal{P}_Y)$  be a complement solution associated with  $i$ . Conditions (a) and (b) of both definitions guarantee that there exists a subset  $\mathbf{VC}$  of  $X_{\downarrow \text{cc}(X \setminus S)} \cup Y_{\downarrow \mathcal{P}_Y}$  such that there is a one-to-one correspondence between the blocks of  $\mathbf{VC}$  and the representatives in  $\mathcal{X}_{\overline{vc}}^S \cup \mathcal{X}_{\overline{vc}}^S \cup \mathcal{Y}_{\overline{vc}}^S \cup \mathcal{Y}_{\overline{vc}}^S$ .

Condition (c) of Definition 12 guarantees that the connections between  $X_{\downarrow \text{cc}(X \setminus S)}$  and  $\mathbf{VC}$  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. In particular, by Fact 7, Conditions (c), (d) and (e) together imply that  $G[X]$  and  $G[Y]$  are  $S$ -forests.

Finally, as explained earlier, the last conditions of both definitions ensure that  $\mathbf{VC}$  the set described by  $\mathcal{X}_{\overline{vc}}^S, \mathcal{X}_{\overline{vc}}^S, \mathcal{Y}_{\overline{vc}}^S$  and  $\mathcal{Y}_{\overline{vc}}^S$  is a vertex cover of  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$ . Notice that  $X \setminus V(\mathbf{VC}_X)$  and  $V(\mathcal{Y}_{\overline{vc}})$  correspond the set of vertices in  $X$  and  $Y$ , respectively, that do not belong to a block in the vertex cover  $\mathbf{VC}$ . Such observations are used to prove the following two results.

**Lemma 14** Let  $X \subseteq V_x$  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)  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  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** By Lemma 9, there exists an  $\overline{S}$ -contraction  $\mathcal{P}_Y$  of  $Y$  such that the following properties are satisfied:

- (A)  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  is a forest,
- (B) for all  $P \in \text{cc}(X \setminus S) \cup \mathcal{P}_Y$  and all  $v \in (X \cup Y) \cap S$ , we have  $|N(v) \cap P| \leq 1$ ,
- (C) the graph  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  admits a vertex cover  $\mathbf{VC}$  of size at most  $4\text{mim}(V_x)$  such that the neighborhoods of the blocks in  $\mathbf{VC}$  are pairwise distinct in  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$ .

We construct  $i = (\mathcal{X}_{\text{vc}}^{\bar{S}}, \mathcal{X}_{\text{vc}}^S, \mathcal{X}_{\bar{\text{vc}}}, \mathcal{Y}_{\text{vc}}^{\bar{S}}, \mathcal{Y}_{\text{vc}}^S) \in \mathbb{I}_x$  from VC as follows:

- $\mathcal{X}_{\text{vc}}^S = \{\text{rep}_1^{V_x}(\{v\}) \mid \{v\} \in (X \cap S) \cap \text{VC}\},$
- $\mathcal{X}_{\text{vc}}^{\bar{S}} = \{\text{rep}_2^{V_x}(P) \mid P \in \text{cc}(X \setminus S) \cap \text{VC}\},$
- $\mathcal{X}_{\bar{\text{vc}}} = \text{rep}_1^{V_x}(X \setminus V(\text{VC})),$
- $\mathcal{Y}_{\text{vc}}^{\bar{S}} = \{\text{rep}_2^{V_x}(P) \mid P \in \mathcal{P}_Y \cap \text{VC}\},$
- $\mathcal{Y}_{\text{vc}}^S = \{\text{rep}_1^{V_x}(\{v\}) \mid \{v\} \in (Y \cap S) \cap \text{VC}\}.$

Since  $|\text{VC}| \leq 4\text{mim}(V_x)$ , we have  $|\mathcal{X}_{\text{vc}}^{\bar{S}}| + |\mathcal{X}_{\text{vc}}^S| + |\mathcal{Y}_{\text{vc}}^{\bar{S}}| + |\mathcal{Y}_{\text{vc}}^S| \leq 4\text{mim}(V_x)$  and thus we have  $i \in \mathbb{I}_x$ .

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 Property (C), i.e. the neighborhoods of the blocks in VC are pairwise distinct in  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$ . So, the blocks in  $X_{\downarrow \text{cc}(X \setminus S)} \cap \text{VC}$  are pairwise non-equivalent for the  $d$ -neighbor equivalence over  $V_x$  for any  $d \in \mathbb{N}^+$  including 1 and 2. Consequently, there is a one to one correspondence between the blocks of  $X_{\downarrow \text{cc}(X \setminus S)} \cap \text{VC}$  and the representatives in  $\mathcal{X}_{\text{vc}}^{\bar{S}} \cup \mathcal{X}_{\text{vc}}^S$ .

It remains to prove Conditions (c), (d) and (e). We claim that Condition (c) is satisfied:  $\text{aux}(X, i)$  is a forest. Observe that, by construction,  $\text{aux}(X, i)$  is isomorphic to the graph  $G[X_{\downarrow \text{cc}(X \setminus S)} \mid Y_{\downarrow \mathcal{P}_Y} \cap \text{VC}]$ . Indeed, for every  $P \in Y_{\downarrow \mathcal{P}_Y} \cap \text{VC}$ , by construction, there exists a unique  $U \in \mathcal{Y}_{\text{vc}}^{\bar{S}} \cup \mathcal{Y}_{\text{vc}}^S$  such that  $U \equiv_1^{\bar{V}_x} P$  or  $U \equiv_2^{\bar{V}_x} P$ . In both case, we have  $N(U) \cap V_x = N(P) \cap V_x$  and thus, the neighborhood of  $P$  in  $G[X_{\downarrow \text{cc}(X \setminus S)} \mid Y_{\downarrow \mathcal{P}_Y} \cap \text{VC}]$  is the same as the neighborhood of  $U$  in  $\text{aux}(X, i)$ . Since  $G[X_{\downarrow \text{cc}(X \setminus S)} \mid Y_{\downarrow \mathcal{P}_Y} \cap \text{VC}]$  is a subgraph of  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  and this latter graph is a forest, we deduce that  $\text{aux}(X, i)$  is also a forest.

We deduce that Conditions (d) and (e) are satisfied from property (B) and Fact 11. Consequently,  $X$  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 argument used earlier, we deduce that Conditions (a) and (b) of Definition 13 are satisfied. By Property (A)  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  is a forest, as a subgraph,  $G[Y]_{\downarrow \mathcal{P}_Y}$  is also a forest and thus Condition (c) is satisfied. Conditions (d) and (e) are satisfied from Property (B) and Fact 11.

It remains to prove Condition (f):  $N(\mathcal{X}_{\bar{\text{vc}}}) \cap \mathcal{Y}_{\bar{\text{vc}}} = \emptyset$  where  $\mathcal{Y}_{\bar{\text{vc}}}$  contains the blocks  $\{v\} \in (Y \cap S)_1$  such that  $\text{rep}_1^{\bar{V}_x}(\{v\}) \notin \mathcal{Y}_{\text{vc}}^S$  and the blocks  $P \in \mathcal{P}_Y$  such that  $\text{rep}_2^{\bar{V}_x}(P) \notin \mathcal{Y}_{\text{vc}}^{\bar{S}}$ . By construction,  $\mathcal{Y}_{\bar{\text{vc}}} = Y \setminus V(\text{VC})$ . Since, VC is a vertex cover of the graph  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$ , there are no edges between  $X_{\downarrow \text{cc}(X \setminus S)} \setminus \text{VC}$  and  $Y_{\downarrow \mathcal{P}_Y} \setminus \text{VC}$  in  $G[X, Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$ . We deduce that  $N(X \setminus V(\text{VC})) \cap \mathcal{Y}_{\bar{\text{vc}}} = \emptyset$ . Since  $X \setminus V(\text{VC}) \equiv_1^{V_x} \mathcal{X}_{\bar{\text{vc}}}$ , we conclude that  $N(\mathcal{X}_{\bar{\text{vc}}}) \cap \mathcal{Y}_{\bar{\text{vc}}} = \emptyset$ . This proves that  $(Y, \mathcal{P}_Y)$  is a complement solution associated with  $i$ . □

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



**Proof** Assume that  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  is a forest. By Fact 7, in order to prove that  $G[X \cup Y]$  is an  $S$ -forest, it is enough to prove that for all  $v \in (X \cup Y) \cap S$  and all  $P \in cc(X \setminus S) \cup \mathcal{P}_Y$ , we have  $|N(v) \cap P| \leq 1$ . Let us prove this statement for a vertex  $v \in X \cap S$  the proof is symmetric for  $v \in Y \cap S$ . Let  $P \in (X \cup Y)_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ . If  $P \notin (cc(X \setminus S) \cup \mathcal{P}_Y)$ , then  $P$  is a singleton in  $\binom{X \cup Y}{1}$  and we are done. If  $P \in cc(X \setminus S)$ , then Condition (e) of Definition 12 guarantees that we have  $|N(v) \cap P| \leq 1$ .

Assume now that  $P \in \mathcal{P}_Y$ . Suppose first that  $\text{rep}_2^{\overline{V_x}}(P) \notin \mathcal{Y}_{\overline{V_c}}^S$ . From Condition (f) of Definition 13, we have  $N(P) \cap \mathcal{X}_{\overline{V_c}} = \emptyset$ . Let  $r = \text{rep}_1^{V_x}(\{v\})$ . From the definition of  $\mathcal{X}_{\overline{V_c}}$  in Definition 12, we deduce that if  $r \notin \mathcal{X}_{\overline{V_c}}^S$ , then  $N(v) \cap \overline{V_x} \subseteq N(\mathcal{X}_{\overline{V_c}})$  and thus  $N(v) \cap P = \emptyset$ . On the other hand, if  $r \in \mathcal{X}_{\overline{V_c}}^S$ , then Condition (d) of Definition 13 ensures that  $|N(r) \cap P| \leq 1$  and thus  $|N(v) \cap P| \leq 1$ . Now, suppose that  $\text{rep}_2^{\overline{V_x}}(P) \in \mathcal{Y}_{\overline{V_c}}^S$ . By Condition (e) of Definition 12, we know that  $|N(v) \cap \text{rep}_2^{\overline{V_x}}(P)| \leq 1$ . From Fact 11, we conclude that  $|N(v) \cap P| \leq 1$ . This concludes the proof of Lemma 15.  $\square$

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  $|\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)}$  because  $\sim_i$  generates at most  $(4\text{mim}(V_x))^{4\text{mim}(V_x)}$  equivalence classes.

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

**Definition 16** ( $cc(X, i)$ ) Let  $i = (\mathcal{X}_{\overline{V_c}}^S, \mathcal{X}_{\overline{V_c}}^S, \mathcal{X}_{\overline{V_c}}^S, \mathcal{Y}_{\overline{V_c}}^S, \mathcal{Y}_{\overline{V_c}}^S) \in \mathbb{I}_x$  and  $X \subseteq V_x$  be a partial solution associated with  $i$ . For each connected component  $C$  of  $\text{aux}(X, i)$ , we define the set  $C_{V_c}$  as follows:

- for every  $\overline{U} \in C$  such that  $U \in \mathcal{Y}_{\overline{V_c}}^S \cup \mathcal{Y}_{\overline{V_c}}^S$ , we have  $U \in C_{V_c}$ ,
- for every  $\{v\} \in \binom{X \cap S}{1} \cap C$  such that  $\{v\} \equiv_1^{V_x} R$  for some  $R \in \mathcal{X}_{\overline{V_c}}^S$ , we have  $R \in C_{V_c}$ ,
- for every  $U \in cc(X \setminus S)$  such that  $U \equiv_2^{V_x} R$  for some  $R \in \mathcal{X}_{\overline{V_c}}^S$ , we have  $R \in C_{V_c}$ .

We define  $cc(X, i)$  as the collection  $\{C_{V_c} \mid C \text{ is a connected component of } \text{aux}(X, i)\}$ .

For a connected component  $C$  of  $\text{aux}(X, i)$ , the set  $C_{V_c}$  contains  $C \cap (\mathcal{Y}_{\overline{V_c}}^S \cup \mathcal{Y}_{\overline{V_c}}^S)$  and the representatives of the blocks in  $C \cap X_{\downarrow cc(X \setminus S)} \cap VC$  with VC the vertex cover described by  $i$ . Consequently, for every  $X \subseteq V_x$  and  $i \in \mathbb{I}_x$ , the collection  $cc(X, i)$  is partition of  $\mathcal{X}_{\overline{V_c}}^S \cup \mathcal{X}_{\overline{V_c}}^S \cup \mathcal{Y}_{\overline{V_c}}^S \cup \mathcal{Y}_{\overline{V_c}}^S$ . For the example given in Fig. 3, observe that  $cc(X, i)$  is the partition that contains  $\{R_1, U_1, U_2\}$ ,  $\{R_2, U_3\}$  and  $\{U_4\}$  (see also Fig. 13)

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  $cc(X, i) = cc(W, i)$ . 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. Figure 14 gives an example of two  $i$ -equivalent partial solutions.

**Lemma 17** *Let  $i = (\mathcal{X}_{vc}^{\bar{S}}, \mathcal{X}_{vc}^S, \mathcal{X}_{\bar{vc}}, \mathcal{Y}_{vc}^{\bar{S}}, \mathcal{Y}_{vc}^S) \in \mathbb{I}_x$ . 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  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  is a forest if and only if the graph  $G[W \cup Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$  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  $i$ . To prove this lemma, we show that if  $G[W \cup Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$  contains a cycle, then  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  contains a cycle too. See Fig. 16 for some intuitions on this proof. We will use the following notation in this proof.

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

- all  $\{v\} \in \binom{Z \cap S}{1}$  such that  $rep_1^{V_x}(\{v\}) \in \mathcal{X}_{vc}^S$ ,
- all  $P \in cc(Z \setminus S)$  such that  $rep_2^{V_x}(P) \in \mathcal{X}_{vc}^{\bar{S}}$ .

We define also  $VC_Y$  as the set that contains:

- all  $\{v\} \in \binom{Y \cap S}{1}$  such that  $rep_1^{\bar{V}_x}(\{v\}) \in \mathcal{Y}_{vc}^S$ ,
- all  $P \in \mathcal{P}_Y$  such that  $rep_2^{\bar{V}_x}(P) \in \mathcal{Y}_{vc}^{\bar{S}}$ .

The sets  $VC_X, VC_W$  and  $VC_Y$  contain the blocks in  $X_{\downarrow cc(X \setminus S)}, W_{\downarrow cc(W \setminus S)}$  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 edge-disjoint subgraphs of  $G[Z \cup Y]_{\downarrow cc(Z \setminus S) \cup \mathcal{P}_Y}$ :

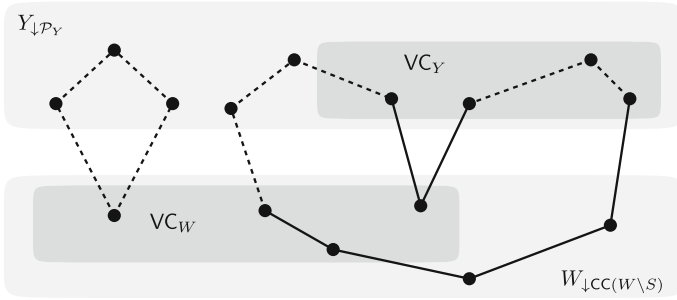
- $\overline{G_Z} = G[Z_{\downarrow cc(Z \setminus S)} \mid VC_Y]$ ,
- $\underline{G_Z} = G[Z \cup Y]_{\downarrow cc(Z \setminus S) \cup \mathcal{P}_Y} - \overline{G_Z}$ .

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

**Fact 18** *For any  $Z \in \{X, W\}$ , the set  $VC_Z \cup VC_Y$  is a vertex cover of  $G[Z, Y]_{\downarrow cc(Z \setminus S) \cup \mathcal{P}_Y}$ .*

**Proof** First observe that  $N(Y \setminus V(VC_Y)) \cap \mathcal{X}_{\bar{vc}} = \emptyset$  thanks to Condition (f) of Definition 13. Moreover, we have  $\mathcal{X}_{\bar{vc}} \equiv_1^{V_x} Z \setminus V(VC_Z)$  by Condition (f) of Definition 12. We conclude that there are no edges between  $Y_{\downarrow \mathcal{P}_Y} \setminus VC_Y$  and  $Z_{\downarrow cc(Z \setminus S)} \setminus VC_Z$  in  $G[Z, Y]_{\downarrow cc(Z \setminus S) \cup \mathcal{P}_Y}$ . Hence,  $VC_Z \cup VC_Y$  is a vertex cover of  $G[Z, Y]_{\downarrow cc(Z \setminus S) \cup \mathcal{P}_Y}$ .  $\square$

Assume that  $G[W \cup Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$  contains a cycle  $C$ . Our task is to show that  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  contains a cycle as well. We first explore properties of  $C$  with



**Fig. 4** How cycles in  $G[W \cup Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$  may interact with the graphs  $G_W$  and  $\overline{G_W}$ . The solid edges belong to  $G_W$  and the dashed edges belong to  $\overline{G_W}$

respect to  $G_W$  and  $\overline{G_W}$ . Since the graph  $\text{aux}(W, i)$  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 through a block of  $W_{\downarrow cc(W \setminus S)}$  because  $G[Y]_{\downarrow \mathcal{P}_Y}$  is a forest. Consequently, (and because from Fact 18  $VC_W \cup VC_Y$  is a vertex cover of  $G[W \cup Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$ ), we deduce that  $C$  is the concatenation of edge-disjoint paths  $P_1 \dots P_t$  such that for each  $\ell \in [t]$  we have:

- $P_\ell$  is a non-empty path with endpoints in  $VC_W \cup VC_Y$  and internal blocks not in  $VC_W \cup VC_Y$  and  $P_\ell$  is either 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  $\overline{G_W}$ .

Given an endpoint  $U \in VC_W \cup VC_Y$  of one of the paths  $P_1, \dots, P_t$ , we define  $U_X$  and  $U_i$  as the analogs of  $U$  in  $VC_X \cup VC_Y$  and  $\mathcal{X}_{vc}^S \cup \mathcal{X}_{vc}^S \cup \mathcal{Y}_{vc}^S \cup \mathcal{Y}_{vc}^S$ , respectively, as follows:

- if  $U \in cc(W \setminus S)$ , then  $U_X$  and  $U_i$  are the unique elements of  $cc(X \setminus S)$  and  $\mathcal{X}_{vc}^S$ , respectively, such that  $U \equiv_2^{V_x} U_X \equiv_2^{V_x} U_i$ ,
- 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  $\mathcal{X}_{vc}^S$ , 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  $\mathcal{Y}_{vc}^S$  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  $\mathcal{Y}_{vc}^S$  such that  $U \equiv_1^{\overline{V_x}} U_i$ .

Observe that  $U_X$  and  $U_i$  exist by Conditions (a) and (b) of Definitions 12 and 13.

For each  $\ell \in [t]$ , we construct a non-empty path  $P'_\ell$  whose endpoints are the analogs in  $VC_X \cup VC_Y$  of the endpoints of  $P_\ell$  and such that if  $P_\ell$  is a path in  $G_W$  (resp.  $\overline{G_W}$ ), then  $P'_\ell$  is a path in  $G_X$  (resp.  $\overline{G_X}$ ). This is sufficient to prove the claim because thanks to this, we can construct a closed walk in  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  by concatenating the paths  $P'_1, \dots, P'_t$ . Since  $G_X$  and  $\overline{G_X}$  are edge-disjoint and the paths  $P'_1, \dots, P'_t$  are non-empty paths, this closed walk must contain a cycle.

Let  $\ell \in [t]$  and  $U, T$  be the endpoints of  $P_\ell$ . We denote by  $U_X, T_X, U_i$  and  $T_i$  the analogs of  $U$  and  $T$  in  $VC_X \cup VC_Y$  and  $\mathcal{X}_{vc}^S \cup \mathcal{X}_{vc}^S \cup \mathcal{Y}_{vc}^S \cup \mathcal{Y}_{vc}^S$ , respectively.

First, assume that  $P_\ell$  is path of  $G_W$ . Observe that  $U_i$  and  $T_i$  belong to the same partition class of  $cc(W, i)$ . This follows from the definitions of  $U_i, T_i$  and the fact that  $G_W$  is isomorphic to  $aux(W, i)$ . As  $W \sim_i X$ , we deduce that  $U_i$  and  $T_i$  belong to the same partition class of  $cc(X, i)$ . By construction,  $U_i$  and  $T_i$  are the analogs of  $U_X$  and  $T_X$  in  $X_{VC}^S \cup X_{VC}^S \cup Y_{VC}^S \cup Y_{VC}^S$ . We conclude that  $U_X$  and  $T_X$  are connected in  $G_X$  via a path  $P'_\ell$ . We claim that  $P'_\ell$  is not empty because we have  $U_X \neq T_X$ . As  $P_\ell$  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  $P_\ell$  is a non-empty path of  $\overline{G_W}$ . Since  $VC_W \cup VC_Y$  is a vertex cover of  $G[W, Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$ , the blocks in  $W_{\downarrow cc(W \setminus S)}$  that do not belong to  $VC_W$  are isolated in  $\overline{G_W}$ . As  $P_\ell$  is not empty, the blocks of  $P_\ell$  which belong to  $W_{\downarrow cc(W \setminus S)}$  are in  $VC_W$ . Because the internal blocks of the paths  $P_1, \dots, P_t$  are not in  $VC_W \cup VC_Y$ , we deduce that the internal blocks of  $P_\ell$  belong to  $Y_{\downarrow \mathcal{P}_Y}$ . We distinguish the following cases:

- If both endpoints of  $P_\ell$  belong to  $VC_Y$ , then  $U_X = U, T_X = T$  and all the blocks of  $P_\ell$  belong to  $Y_{\downarrow \mathcal{P}_Y}$ . It follows that  $P_\ell$  is a non-empty path of  $\overline{G_X}$  because  $G[Y]_{\downarrow \mathcal{P}_Y}$  is a subgraph of  $\overline{G_X}$ . In this case, we take  $P'_\ell = P_\ell$ .
- Assume now that one or two endpoints of  $P_\ell$  belong to  $VC_W$ . Suppose w.l.o.g. that  $U$  belongs to  $VC_W$ . Since  $P_\ell$  is non-empty and the internal blocks of  $P_\ell$  are in  $Y_{\downarrow \mathcal{P}_Y}$ ,  $U$  has a neighbor  $Q \in Y_{\downarrow \mathcal{P}_Y}$  in  $P_\ell$ . We claim that  $Q$  is adjacent to  $U_X$  in  $\overline{G_X}$ . By definition of  $U_X$ , we have  $U \equiv_d^{V_x} U_X$  for some  $d \in \{1, 2\}$  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  $N(U) \cap Q \neq \emptyset$ . It follows that  $N(U_X) \cap Q \neq \emptyset$  and thus  $Q$  and  $U_X$  are adjacent in  $\overline{G_X}$ . Symmetrically, we can prove that if  $T \in VC_W$ , then the neighbor of  $T$  in  $P_\ell$  is adjacent to  $T_X$  in  $\overline{G_X}$ .

Hence, the neighbors of  $U$  and  $T$  in  $P_\ell$  are adjacent to  $U_X$  and  $T_X$  respectively in  $\overline{G_X}$ . We obtain  $P'_\ell$  from  $P_\ell$  by replacing  $U$  and  $T$  by  $U_X$  and  $T_X$ . Since the internal blocks of  $P_\ell$  belong to  $Y_{\downarrow \mathcal{P}_Y}$  and  $G[Y]_{\downarrow \mathcal{P}_Y}$  is a subgraph of  $\overline{G_X}$ , we deduce that  $P'_\ell$  is a path of  $\overline{G_X}$ . The path  $P'_\ell$  is not-empty because it contains  $U_X$  and  $Q$  which are distinct blocks of  $\overline{G_X}$  as  $U_X \in VC_X$  (since  $U \in VC_W$  by assumption) and  $Q \in Y_{\downarrow \mathcal{P}_Y}$ .

□

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  $\mathcal{A}$  are as good as the ones we obtain from  $\mathcal{B}$ . Firstly, we formalize this notion of representativity.

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

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

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  $\text{best}(\mathcal{A}, Y) = \text{best}(\mathcal{B}, Y)$ .

We recall that  $s\text{-nec}_2(A) = \max(\text{nec}_2(A), \text{nec}_2(\overline{A}))$ .

**Lemma 20** *There exists an algorithm reduce that, given a set  $\mathcal{A} \subseteq 2^{V_x}$ , outputs in time  $O(|\mathcal{A}| \cdot |\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)} \cdot \log(\text{s-nec}_2(V_x)) \cdot n^3)$  a subset  $\mathcal{B} \subseteq \mathcal{A}$  such that  $\mathcal{B}$  represents  $\mathcal{A}$  and  $|\mathcal{B}| \leq |\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)}$ .*

**Proof** Given  $\mathcal{A} \subseteq 2^{V_x}$  and  $i \in \mathbb{I}_x$ , we define  $\text{reduce}(\mathcal{A}, i)$  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  $w(X)$  is maximum. Moreover, we define  $\text{reduce}(\mathcal{A}) = \bigcup_{i \in \mathbb{I}_x} \text{reduce}(\mathcal{A}, i)$ .

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

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  $w(X) = \text{best}(\mathcal{A}, Y)$ . By Lemma 14, there exists  $i \in \mathbb{I}_x$  and an  $\overline{S}$ -contraction  $\mathcal{P}_Y$  of  $Y$  such that (1)  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  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  $\text{reduce}(\mathcal{A}, i)$ , there exists  $W \in \text{reduce}(\mathcal{A})$  such that  $W$  is a partial solution associated with  $i$ ,  $X \sim_i W$ , and  $w(W) \geq w(X)$ . By Lemma 17 and since  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  is a forest, we deduce that  $G[W \cup Y]_{\downarrow \text{cc}(W \setminus S) \cup \mathcal{P}_Y}$  is a forest too. Thanks to Lemma 15, we deduce that  $G[W \cup Y]$  is an  $S$ -forest. As  $w(W) \geq w(X) = \text{best}(\mathcal{A}, Y)$ , we conclude that  $\text{best}(\mathcal{A}, Y) = \text{best}(\text{reduce}(\mathcal{A}), Y)$ . Hence,  $\text{reduce}(\mathcal{A})$  represents  $\mathcal{A}$ .

We claim that  $|\text{reduce}(\mathcal{A})| \leq |\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)}$ . For every  $i \in \mathbb{I}_x$  and partial solution  $X$  associated with  $i$ ,  $\text{cc}(X, i)$  is a partition of  $\mathcal{X}_{\overline{V_c}}^{\overline{S}} \cup \mathcal{X}_{\overline{V_c}}^S \cup \mathcal{X}_{\overline{V_c}}^{\overline{S}} \cup \mathcal{Y}_{\overline{V_c}}^{\overline{S}} \cup \mathcal{Y}_{\overline{V_c}}^S$ . Since  $|\mathcal{X}_{\overline{V_c}}^{\overline{S}}| + |\mathcal{X}_{\overline{V_c}}^S| + |\mathcal{Y}_{\overline{V_c}}^{\overline{S}}| + |\mathcal{Y}_{\overline{V_c}}^S| \leq 4\text{mim}(V_x)$ , there are at most  $(4\text{mim}(V_x))^{4\text{mim}(V_x)}$  possible values for  $\text{cc}(X, i)$ . We deduce that, for every  $i \in \mathbb{I}_x$ , the relation  $\sim_i$  generates at most  $(4\text{mim}(V_x))^{4\text{mim}(V_x)}$  equivalence classes, so  $|\text{reduce}(\mathcal{A}, i)| \leq (4\text{mim}(V_x))^{4\text{mim}(V_x)}$  for every  $i \in \mathbb{I}_x$ . By construction, we conclude that  $|\text{reduce}(\mathcal{A})| \leq |\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)}$ .

It remains to prove the runtime. As  $\text{nec}_1(V_x) \leq \text{nec}_2(V_x)$ , by Lemma 5 we can compute in time  $O(\text{s-nec}_2(V_x) \cdot \log(\text{s-nec}_2(V_x)) \cdot n^2)$  the sets  $\mathcal{R}_1^{V_x}, \mathcal{R}_2^{V_x}, \mathcal{R}_2^{\overline{V_x}}$  and data structures which compute  $\text{rep}_1^{V_x}, \text{rep}_2^{V_x}$  and  $\text{rep}_2^{\overline{V_x}}$  in time  $O(\log(\text{s-nec}_2(V_x)) \cdot n^2)$ . 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  $O(|\mathbb{I}_x| \cdot n^2)$ . Since,  $\text{s-nec}_2(V_x) \leq |\mathbb{I}_x|$ , the time required to compute these sets and data structures is less than  $O(|\mathcal{A}| \cdot |\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)} \cdot \log(\text{s-nec}_2(V_x)) \cdot n^3)$ .

For each  $i \in \mathbb{I}_x$  and  $X \in \mathcal{A}$ , we can decide whether  $X$  is a partial solution associated with  $i$  and compute  $\text{aux}(X, i), \text{cc}(X, i)$  in time  $O(\log(\text{s-nec}_2(V_x)) \cdot n^3)$ . For doing so, we simply start by computing  $\text{rep}_2^{V_x}(C)$  and  $\text{rep}_1^{V_x}(\{v\})$  for each  $C \in \text{cc}(X \setminus S)$  and  $\{v\} \in \binom{X \cap S}{1}$ , this is doable in  $O(\log(\text{s-nec}_2(V_x)) \cdot n^3)$  since  $|\text{cc}(X \setminus S)| + |X \cap S| \leq n$ . Then with standard algorithmic techniques, we check whether  $X$  satisfies all the conditions of Definition 12 and compute  $\text{aux}(X, i)$  and  $\text{cc}(X, i)$ .

Given two partial solutions  $X, W$  associated with  $i$ ,  $\text{cc}(X, i)$  and  $\text{cc}(W, i)$ , we can decide whether  $X \sim_i W$  in time  $O(\text{mim}(V_x)) \leq O(n)$ . We deduce that, for

each  $i \in \mathbb{I}_x$ , we can compute  $\text{reduce}(\mathcal{A}, i)$  in time  $O(|\mathcal{A}| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)} \cdot \log(\text{s-nec}_2(V_x)) \cdot n^3)$ . We deduce the running time to compute  $\text{reduce}(\mathcal{A})$  by multiplying the running time needed for  $\text{reduce}(\mathcal{A}, i)$  by  $|\mathbb{I}_x|$ .  $\square$

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  $\mathcal{A} \otimes \mathcal{B}$ , as  $\mathcal{A} \otimes \mathcal{B} := \{X \cup Y \mid X \in \mathcal{A} \text{ and } Y \in \mathcal{B}\}$ . Observe that  $\mathcal{A} \otimes \mathcal{B} = \emptyset$  whenever  $\mathcal{A} = \emptyset$  or  $\mathcal{B} = \emptyset$ .

**Theorem 21** *There exists an algorithm that, given an  $n$ -vertex graph  $G$  and a rooted layout  $(T, \delta)$  of  $G$ , solves SUBSET FEEDBACK VERTEX SET in time*

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

**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  $|\mathcal{A}_x| \leq |\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)}$ . For the leaves  $x$  of  $T$  such that  $V_x = \{v\}$ , we simply take  $\mathcal{A}_x = 2^{V_x} = \{\emptyset, \{v\}\}$ . 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  $\mathcal{A}_x = \text{reduce}(\mathcal{A}_a \otimes \mathcal{A}_b)$ . Once the the set  $\mathcal{A}_r$  is computed with  $r$  the root of  $T$ , our algorithm outputs a set  $X \in \mathcal{A}_r$  of maximum weight.

By Lemma 20, we have  $|\mathcal{A}_x| \leq |\mathbb{I}_x| \cdot (4\text{mim}(V_x))^{4\text{mim}(V_x)}$ , 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  $\text{reduce}(\mathcal{A}_a \otimes \mathcal{A}_b)$  represents  $2^{V_x}$ .*

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

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

As  $\mathcal{A}_a$  represents  $2^{V_a}$ , we have  $\text{best}(\mathcal{A}_a, W \cup Y) = \text{best}(2^{V_a}, W \cup Y)$  and we deduce that  $\text{best}(\mathcal{A}_a \otimes \mathcal{A}_b, Y) = \text{best}(2^{V_a} \otimes \mathcal{A}_b, Y)$ . Symmetrically, as  $\mathcal{A}_b$  represents  $2^{V_b}$ , we infer that  $\text{best}(2^{V_a} \otimes \mathcal{A}_b, Y) = \text{best}(2^{V_a} \otimes 2^{V_b}, Y)$ . Since  $2^{V_a} \otimes 2^{V_b} = 2^{V_x}$ , we conclude that  $\text{best}(\mathcal{A}_a \otimes \mathcal{A}_b, Y)$  equals  $\text{best}(2^{V_x}, Y)$ . As this holds for every  $Y$ , it proves that  $\mathcal{A}_a \otimes \mathcal{A}_b$  represents  $2^{V_x}$ . By Lemma 20, we know that  $\text{reduce}(\mathcal{A}_a \otimes \mathcal{A}_b)$  represents  $\mathcal{A}_a \otimes \mathcal{A}_b$ . As the relation “represents” is transitive, we conclude that  $\text{reduce}(\mathcal{A}_a \otimes \mathcal{A}_b)$  represents  $2^{V_x}$ .  $\square$

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  $T$ . In particular,  $\mathcal{A}_r$  represents  $2^{V(G)}$  with  $r$  the root of  $T$ . By Definition 19,  $\mathcal{A}_r$  contains a set  $X$  of maximum size such that  $G[X]$  is an  $S$ -forest. This proves the correctness of our algorithm.

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  $\mathcal{A}_a \otimes \mathcal{A}_b$  is at most  $|\mathbb{I}_x|^2 \cdot (4\text{mim}(V_x))^{8\text{mim}(V_x)}$  and it can be computed in time  $O(|\mathbb{I}_x|^2 \cdot (4\text{mim}(V_x))^{8\text{mim}(V_x)} \cdot n^2)$ . By Lemma 20, the set  $\mathcal{A}_x = \text{reduce}(\mathcal{A}_a \otimes \mathcal{A}_b)$  is computable in time  $O(|\mathbb{I}_x|^3 \cdot (4\text{mim}(V_x))^{12\text{mim}(V_x)} \cdot \log(s\text{-nec}_2(V_x)) \cdot n^3)$ . This proves the running time.  $\square$

### 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 set of indices with respect to the considered parameters.

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

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

**Proof** For  $A \subseteq V(G)$ , let  $\text{mw}(A)$  be the number of different rows in the matrix  $M_{A, \bar{A}}$ . Observe that, for every  $A \subseteq V(G)$ , we have  $\text{mw}(A) = \{\text{rep}_1^A(\{v\}) \mid v \in A\}$ . Remember that the  $\mathcal{X}_{\text{vc}}^S$ 's and  $\mathcal{Y}_{\text{vc}}^S$ 's are subsets of  $\{\text{rep}_1^{V_x}(\{v\}) \mid v \in V_x\}$  and  $\{\text{rep}_1^{\bar{V}_x}(\{v\}) \mid v \in \bar{V}_x\}$  respectively. From Definition 10, we have  $|\mathcal{X}_{\text{vc}}^{\bar{S}}| + |\mathcal{X}_{\text{vc}}^S| + |\mathcal{Y}_{\text{vc}}^{\bar{S}}| + |\mathcal{Y}_{\text{vc}}^S| \leq 4\text{mim}(V_x)$ , for every  $(\mathcal{X}_{\text{vc}}^{\bar{S}}, \mathcal{X}_{\text{vc}}^S, \mathcal{X}_{\text{vc}}^{\bar{S}}, \mathcal{Y}_{\text{vc}}^{\bar{S}}, \mathcal{Y}_{\text{vc}}^S) \in \mathbb{I}_x$ . Thus, the size of  $\mathbb{I}_x$  is at most

$$\text{nec}_1(V_x) \cdot (\text{nec}_2(V_x) + \text{mw}(V_x) + \text{nec}_2(\bar{V}_x) + \text{mw}(\bar{V}_x))^{4\text{mim}(V_x)}.$$

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

**$\mathbb{Q}$ -Rank-Width** By Lemma 4, we have  $\text{nec}_1(V_x), \text{nec}_2(V_x), \text{nec}_2(\bar{V}_x) \in 2^{O(\text{rw}_{\mathbb{Q}}(V_x) \log(\text{rw}_{\mathbb{Q}}(V_x)))}$ . Moreover, there is at most  $2^{\text{rw}_{\mathbb{Q}}(V_x)}$  different rows in the matrices  $M_{V_x, \bar{V}_x}$  and  $M_{\bar{V}_x, V_x}$ , so  $\text{mw}(V_x)$  and  $\text{mw}(\bar{V}_x)$  are upper bounded by  $2^{\text{rw}_{\mathbb{Q}}(V_x)}$ . By Lemma 3, we have  $4\text{mim}(V_x) \leq 4\text{rw}_{\mathbb{Q}}(V_x)$ . We deduce from these inequalities that

$$|\mathbb{I}_x| \leq 2^{O(\text{rw}_{\mathbb{Q}}(V_x) \log(\text{rw}_{\mathbb{Q}}(V_x)))} \cdot (2^{O(\text{rw}_{\mathbb{Q}}(V_x) \log(\text{rw}_{\mathbb{Q}}(V_x)))} + 2^{\text{rw}_{\mathbb{Q}}(V_x)+1})^{4\text{rw}_{\mathbb{Q}}(V_x)}.$$

We conclude that  $|\mathbb{I}_x| \in 2^{O(\text{rw}_{\mathbb{Q}}(V_x)^2 \log(\text{rw}_{\mathbb{Q}}(V_x)))}$ .

**Mim-Width** By Lemma 4, we know that  $\text{nec}_1(V_x) \leq |V_x|^{\text{mim}(V_x)}$ ,  $\text{nec}_2(V_x) \leq |V_x|^{2\text{mim}(V_x)}$ , and  $\text{nec}_2(\overline{V_x}) \leq |\overline{V_x}|^{2\text{mim}(V_x)}$ . We can assume that  $n > 2$  (otherwise the problem is trivial), so  $\text{nec}_2(V_x) + \text{nec}_2(\overline{V_x}) \leq |V_x|^{2\text{mim}(V_x)} + |\overline{V_x}|^{2\text{mim}(V_x)} \leq n^{2\text{mim}(V_x)}$ . Moreover, notice that, for every  $A \subseteq V(G)$ , we have  $\{\text{rep}_1^A(\{v\}) \mid v \in A\} \leq |A|$ .

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

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

**Theorem 24** *There exist algorithms that solve SUBSET FEEDBACK VERTEX SET in time  $2^{O(\text{rw}(G)^3)} \cdot n^4$  and  $2^{O(\text{rw}_{\mathbb{Q}}(G)^2 \log(\text{rw}_{\mathbb{Q}}(G)))} \cdot n^4$ .*

**Proof** We first compute a rooted layout  $\mathcal{L} = (T, \delta)$  of  $G$  such that  $\text{rw}(\mathcal{L}) \in O(\text{rw}(G))$  or  $\text{rw}_{\mathbb{Q}}(\mathcal{L}) \in O(\text{rw}_{\mathbb{Q}}(G))$ . This is achieved through a  $(3k + 1)$ -approximation algorithm that runs in FPT time  $O(8^k \cdot n^4)$  parameterized by  $k \in \{\text{rw}(G), \text{rw}_{\mathbb{Q}}(G)\}$  [34]. Then, we apply the algorithm given in Theorem 21. Observe that for every node  $x \in V(T)$ , by Lemma 23,  $|\mathbb{I}_x|^3$  lies in  $2^{O(\text{rw}(V_x)^3)}$  and  $2^{O(\text{rw}_{\mathbb{Q}}(V_x)^2 \log(\text{rw}_{\mathbb{Q}}(V_x)))}$  and by Lemma 4,  $s\text{-nec}_2(V_x)$  lies in  $2^{O(\text{rw}(V_x)^2)}$  and  $2^{O(\text{rw}_{\mathbb{Q}}(V_x) \log(\text{rw}_{\mathbb{Q}}(V_x)))}$ . Moreover, Lemma 3 implies that  $\text{mim}(V_x)^{\text{mim}(V_x)}$  is upper bounded by  $2^{\text{rw}(G) \log(\text{rw}(G))}$  and  $2^{\text{rw}_{\mathbb{Q}}(G) \log(\text{rw}_{\mathbb{Q}}(G))}$ . Therefore, we get the claimed runtimes for SFVS since  $T$  contains  $2n - 1$  nodes.  $\square$

Regarding mim-width, our algorithm given below shows that SUBSET FEEDBACK VERTEX SET is in XP parameterized by the mim-width of a given rooted layout. Note that we cannot solve SFVS in FPT time parameterized by the mim-width of a given rooted layout unless  $\text{FPT} = \text{W}[1]$ , since SUBSET FEEDBACK VERTEX SET is known to be  $\text{W}[1]$ -hard for this parameter even for the special case of  $S = V(G)$  [29]. 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, PERMUTATION graphs, CIRCULAR ARC graphs, CONVEX graphs,  $k$ -POLYGON, DILWORTH- $k$  and CO- $k$ -DEGENERATE graphs for 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  $n^{O(\text{mim}(\mathcal{L})^2)}$ .*

**Proof** We apply the algorithm given in Theorem 21. By Lemma 4, we have  $s\text{-nec}_2(V_x) \leq n^{O(\text{mim})}$  and from Lemma 23 we have  $|\mathbb{I}_x|^3 \in n^{O(\text{mim}(V_x)^2)}$ . The claimed runtime for SFVS follows by the fact that the rooted tree  $T$  of  $\mathcal{L}$  contains  $2n - 1$  nodes.  $\square$



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 [20]. In fact, we can solve NODE MULTIWAY CUT by adding a new vertex  $v$  with a large weight that is adjacent to all terminals and, then, run our algorithms for SUBSET FEEDBACK VERTEX SET with  $S = \{v\}$  on the resulting graph. Now observe that any extension of a rooted layout  $\mathcal{L}$  of the original graph to the resulting graph has mim-width  $\text{mim}(\mathcal{L}) + 1$ . Therefore, all of our algorithms given in Theorems 24 and 25 have the same running times for the NODE MULTIWAY CUT problem.

## 4 Conclusion

This paper highlights the importance of the  $d$ -neighbor-equivalence relation to obtain meta-algorithm for several width measures at once. We extend the range of applications of this relation [3, 7, 22, 35] by proving that it is useful for the atypical acyclicity constraint of the SUBSET FEEDBACK VERTEX SET problem. It would be interesting to see whether this relation can be helpful with other kinds of constraints such as 2-connectivity and other generalizations of FEEDBACK VERTEX SET such as the ones studied in [6]. In particular, one could consider the following generalization of ODD CYCLE TRANSVERSAL:

SUBSET ODD CYCLE TRANSVERSAL (SOCT)

**Input:** A graph  $G$  and  $S \subseteq V(G)$ .

**Output:** A set  $X \subseteq V(G)$  of minimum weight such that  $G[\overline{X}]$  does not contain an odd cycle that intersects  $S$ .

Similar to SFVS, we can solve SOCT in time  $k^{O(k)} \cdot n^{O(1)}$  parameterized by treewidth and this is optimal under ETH [2]. We do not know whether SOCT is in XP parameterized by mim-width, though it is in FPT parameterized by clique-width or rank-width, since we can express it in  $\text{MSO}_1$  (with the characterization used in [2]).

For many well-known graph classes a decomposition of bounded mim-width can be found in polynomial time. However, for general graphs it is known that computing mim-width is  $W[1]$ -hard and not in APX unless  $\text{NP} = \text{ZPP}$  [40], while Yamazaki [42] shows that under the small set expansion hypothesis it is not in APX unless  $\text{P} = \text{NP}$ . For dynamic programming algorithms as in this paper, to circumvent the assumption that we are given a decomposition, we want functions  $f, g$  and an algorithm that given a graph of mim-width  $\text{OPT}$  computes an  $f(\text{OPT})$ -approximation to mim-width in time  $n^{g(\text{OPT})}$ , i.e. XP by the natural parameter. This is the main open problem in the field. The first task could be to decide if there is a constant  $c$  and a polynomial-time algorithm that given a graph  $G$  either decides that its mim-width is larger than 1 or else returns a decomposition of mim-width  $c$ .

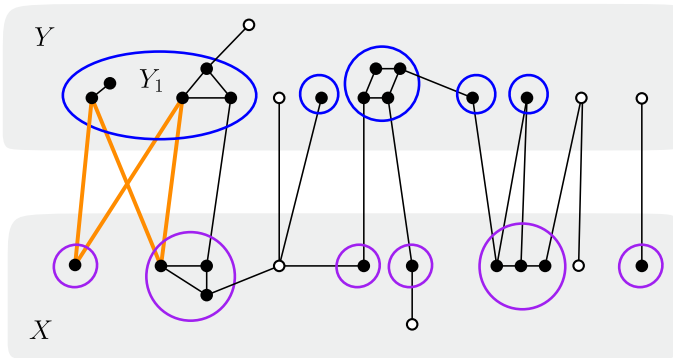
**Funding** Open access funding provided by University of Bergen (incl Haukeland University Hospital).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give

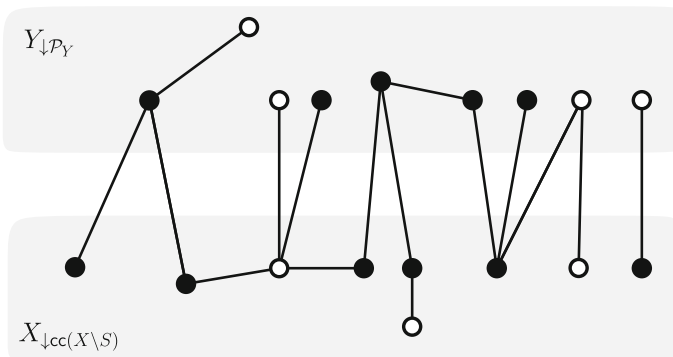
appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### A Explanations with Several Figures

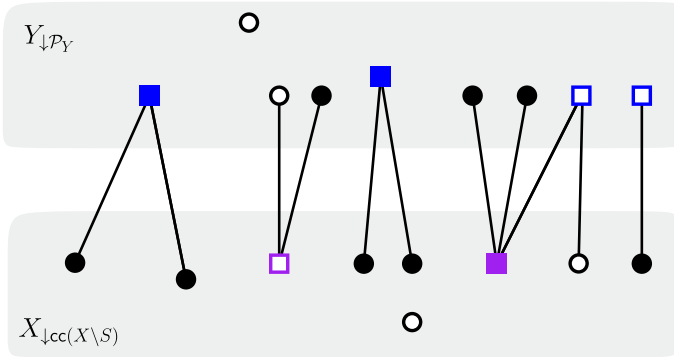
The following figures explain the relation between solutions ( $S$ -forests) and an index  $i$  (Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16).



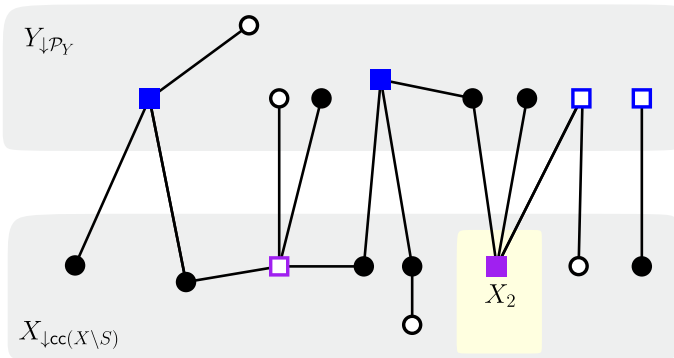
**Fig. 5** Illustration of the  $\bar{S}$ -contraction  $\mathcal{P}_Y$  constructed in Lemma 9. The graph represented is an  $S$ -forest induced by the union of a set  $X \subseteq V_x$  and a set  $Y \subseteq \bar{V}_x$ . The white filled disks represent the vertices in  $S$ . We transform this  $S$ -forest into a forest (see the next figure) by contracting the circled subsets of vertices. The  $\bar{S}$ -contraction used on  $X$  is  $cc(X \setminus S)$ . Observe that  $Y_1$  is the only block of  $\mathcal{P}_Y$  which is not a connected component of  $G[Y \setminus S]$ . We need  $Y_1$  to kill the orange cycle which is not an  $S$ -cycle (Color figure online)



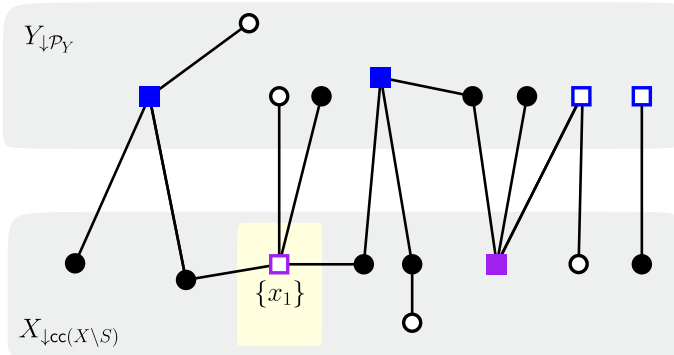
**Fig. 6** The contracted graph  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ . The vertices of this graph are the blocks of  $cc(X \setminus S) \cup \mathcal{P}_Y$  and the singletons  $\{v\}$  for every vertex  $v$  in  $(X \cup Y) \cap S$ , these singletons are represented by the white filled disks



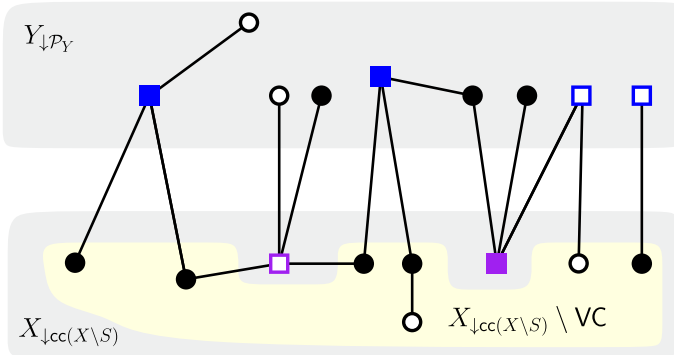
**Fig. 7** The bipartite graph  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ . The squares represent the blocks in a vertex cover VC of  $G[X, Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  that satisfies the properties of Lemma 9



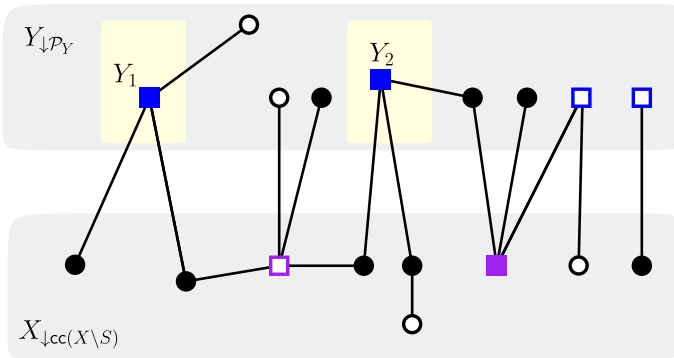
**Fig. 8** The graph  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$ . As done in Lemma 14, we construct from the vertex cover VC an index  $i = (x_{vc}^{\bar{S}}, x_{vc}^S, x_{vc}^{\bar{S}}, y_{vc}^{\bar{S}}, y_{vc}^S) \in \mathbb{I}_x$  such that  $X$  is a partial solution associated with  $i$  and  $(Y, \mathcal{P}_Y)$  is a complement solution associated with  $i$ . The set  $\mathcal{X}_{vc}^{\bar{S}}$  contains the representatives of the blocks in  $cc(X \setminus S)$  that are in the vertex cover VC. In this example, we have  $\mathcal{X}_{vc}^{\bar{S}} = \{R_2\}$  where  $R_2 = \text{rep}_2^{Y_x}(X_2)$



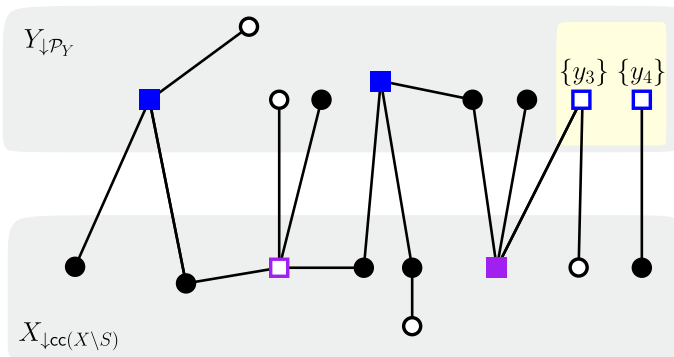
**Fig. 9** The set  $\mathcal{X}_{vc}^S$  contains the representatives of the singletons  $\{x\} \in VC$  with  $x \in S$ . In this example, we have  $\mathcal{X}_{vc}^S = \{R_1\}$  where  $R_1 = \text{rep}_1^{Y_x}(\{x_1\})$



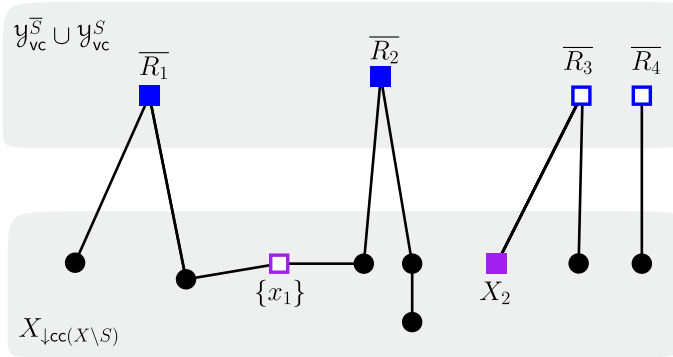
**Fig. 10** The set  $\mathcal{X}_{\overline{VC}} = \text{rep}_1^{V_X}(X \setminus V(VC))$ . It is the representative set of the union of the blocks in  $X_{\downarrow cc(X \setminus S)} \setminus VC$



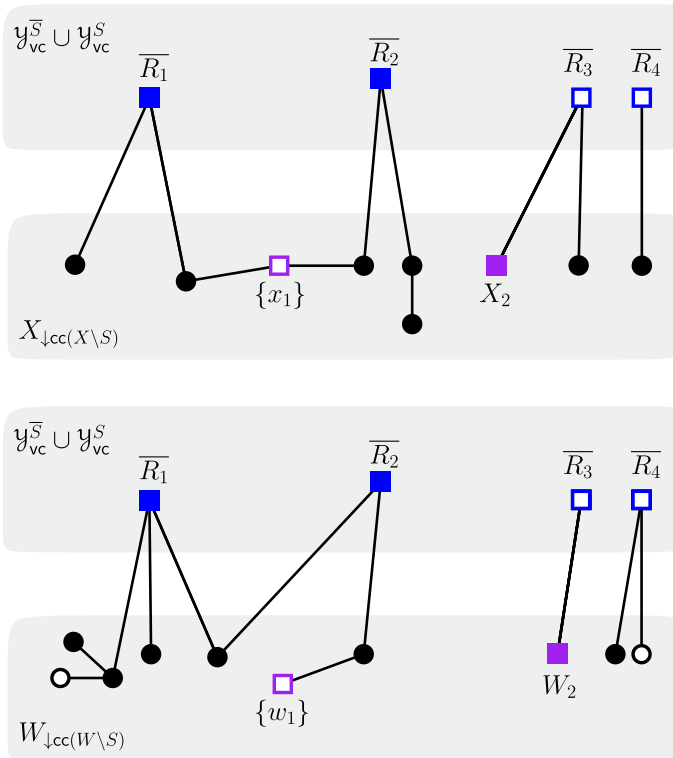
**Fig. 11** The set  $\mathcal{Y}_{\overline{VC}}^{\overline{S}}$  contains the representatives of the blocks in  $\mathcal{P}_Y \cap VC$ . In this example, we have  $\mathcal{Y}_{\overline{VC}}^{\overline{S}} = \{\overline{R_1}, \overline{R_2}\}$  where  $\overline{R_\ell} = \text{rep}_2^{\overline{V_X}}(Y_\ell)$  for  $\ell = 1, 2$



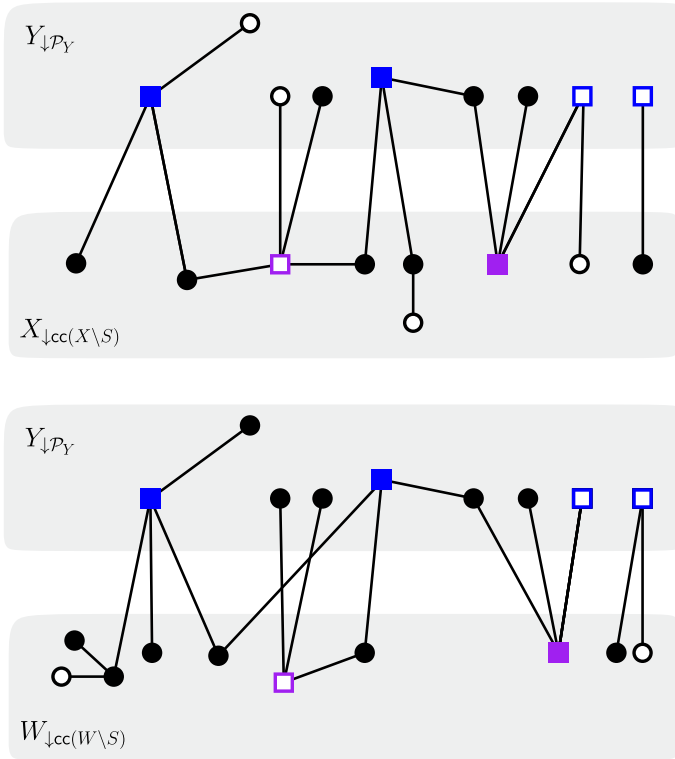
**Fig. 12** The set  $\mathcal{Y}_{\overline{VC}}^S$  contains the representatives of the singletons  $\{y\} \in VC$  with  $y \in S$ . In this example, we have  $\mathcal{Y}_{\overline{VC}}^S = \{\overline{R_3}, \overline{R_4}\}$  where  $\overline{R_\ell} = \text{rep}_1^{\overline{V_X}}(\{y_\ell\})$  for  $\ell = 3, 4$



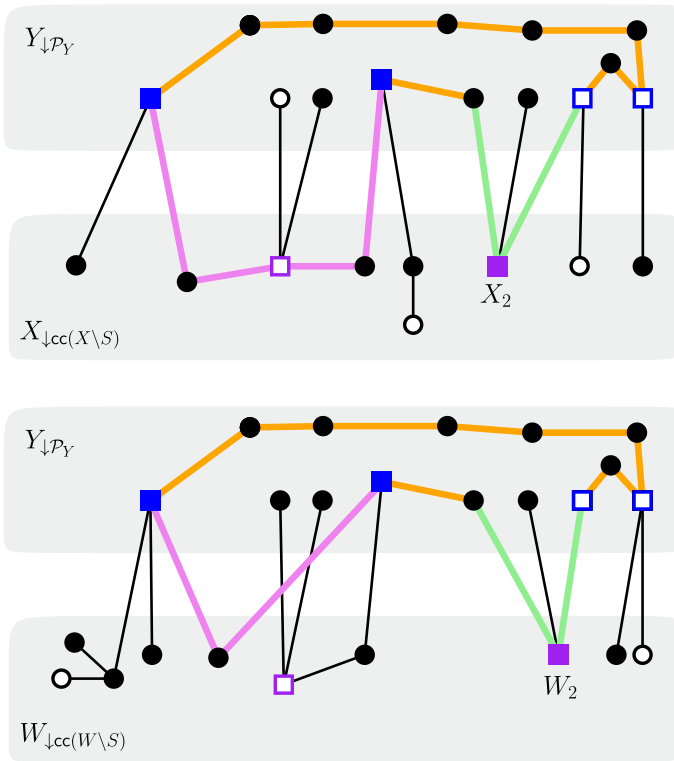
**Fig. 13** The auxiliary graph  $\text{aux}_x(X, i)$ . It can be obtained from  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  by (1) removing the edges between blocks of  $Y_{\downarrow \mathcal{P}_Y}$ , (2) removing the blocks of  $Y_{\downarrow \mathcal{P}_Y}$  that do not belong to VC and (3) replacing each remaining blocks of  $Y_{\downarrow \mathcal{P}_Y}$  by its representatives. The blocks of the partition  $\text{cc}(X, i)$  are  $\{R_1, \overline{R}_1, R_2\}$ ,  $\{R_2, \overline{R}_3\}$  and  $\{\overline{R}_4\}$  where  $R_1$  and  $R_2$  are the representatives of  $\{x_1\}$  and  $X_2$  respectively



**Fig. 14** The auxiliary graphs  $\text{aux}_x(X, i)$  and  $\text{aux}_x(W, i)$  with  $W$  a partial solution associated with  $i$ . We have  $R_1 = \text{rep}_1^{V_x}(\{w_1\}) = \text{rep}_1^{V_x}(\{x_1\})$  and  $R_2 = \text{rep}_2^{V_x}(W_2) = \text{rep}_2^{V_x}(X_2)$ . Observe that  $\text{cc}(X, i) = \text{cc}(W, i) = \{\{R_1, \overline{R}_1, R_2\}, \{R_2, \overline{R}_3\}, \{\overline{R}_4\}\}$ . Consequently,  $X$  and  $W$  are  $i$ -equivalent



**Fig. 15** The graphs  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  and  $G[W \cup Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$ . As  $X$  and  $W$  are  $i$ -equivalent and  $G[X \cup Y]_{\downarrow cc(X \setminus S) \cup \mathcal{P}_Y}$  is a forest, by Lemma 17,  $G[W \cup Y]_{\downarrow cc(W \setminus S) \cup \mathcal{P}_Y}$  is also a forest. Thus, by Lemma 15,  $G[W \cup Y]$  is an  $S$ -forest. In fact, these two lemmas implies that for every complement solution  $(Y', \mathcal{P}'_Y)$  associated with  $i$ ,  $G[X \cup Y']$  is an  $S$ -forest if and only if  $G[W \cup Y']$  is an  $S$ -forest



**Fig. 16** New examples for the graphs  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  and  $G[W \cup Y]_{\downarrow \text{cc}(W \setminus S) \cup \mathcal{P}_Y}$ . To prove Lemma 17, we prove that if  $G[W \cup Y]_{\downarrow \text{cc}(W \setminus S) \cup \mathcal{P}_Y}$  contains a cycle, then  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  contains a cycle. We use the following arguments. The orange paths exist in both graphs because these paths only use blocks in  $Y_{\downarrow \mathcal{P}_Y}$ . Since  $\text{cc}(X, i) = \text{cc}(W, i)$ , the purple path in  $G[W \cup Y]_{\downarrow \text{cc}(W \setminus S) \cup \mathcal{P}_Y}$  implies the existence of the purple path in  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$ . Finally, as  $X_2 \equiv_2^{V_X} W_2$ , the blocks in  $Y_{\downarrow \mathcal{P}_Y}$  adjacent to  $W_2$  are also adjacent to  $X_2$ . Thus, the green path in  $G[W \cup Y]_{\downarrow \text{cc}(W \setminus S) \cup \mathcal{P}_Y}$  implies the existence of the green path in  $G[X \cup Y]_{\downarrow \text{cc}(X \setminus S) \cup \mathcal{P}_Y}$  (Color figure online)

## References

1. Belmonte, R., Vatshelle, M.: Graph classes with structured neighborhoods and algorithmic applications. *Theor. Comput. Sci.* **511**, 54–65 (2013)
2. Bergougnoux, B., Bonnet, É., Brettell, N., Kwon, O.: Close relatives of feedback vertex set without single-exponential algorithms parameterized by treewidth. In: 15th International Symposium on Parameterized and Exact Computation, IPEC 2020, December 14–18, 2020, Hong Kong, China (Virtual Conference), pp. 3:1–3:17 (2020)
3. Bergougnoux, B., Kanté, M.M.: 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, pp. 17:1–17:14 (2019)
4. Bodlaender, H.L.: 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, pp. 1–14 (2006)
5. Bodlaender, H.L., Cygan, M., Kratsch, S., Nederlof, J.: Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inform. Comput.* **243**, 86–111 (2015)

6. Bonnet, É., Brettell, N., Kwon, O., Marx, D.: Generalized feedback vertex set problems on bounded-treewidth graphs: chordality is the key to single-exponential parameterized algorithms. *Algorithmica* **81**(10), 3890–3935 (2019)
7. Bui-Xuan, B.-M., Telle, J.A., Vatshelle, M.: Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theor. Comput. Sci.* **511**, 66–76 (2013)
8. Calinescu, G.: *Multitway Cut*. Springer, Berlin (2008)
9. Chen, J., Liu, Y., Lu, S.: An improved parameterized algorithm for the minimum node multitway cut problem. *Algorithmica* **55**, 1–13 (2009)
10. Chitnis, R.H., Fomin, F.V., Lokshtanov, D., Misra, P., Ramanujan, M.S., Saurabh, S.: Faster exact algorithms for some terminal set problems. *J. Comput. Syst. Sci.* **88**, 195–207 (2017)
11. Corneil, D.G., Fonlupt, J.: The complexity of generalized clique covering. *Discrete Appl. Math.* **22**(2), 109–118 (1988)
12. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.* **33**(2), 125–150 (2000)
13. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. *Discrete Appl. Math.* **101**(1–3), 77–114 (2000)
14. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: On multitway cut parameterized above lower bounds. *TOCT* **5**(1), 3:1–3:11 (2013)
15. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: Subset feedback vertex set is fixed-parameter tractable. *SIAM J. Discrete Math.* **27**(1), 290–309 (2013)
16. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. *SIAM J. Comput.* **23**(4), 864–894 (1994)
17. Even, G., Naor, J., Zosin, L.: An 8-approximation algorithm for the subset feedback vertex set problem. *SIAM J. Comput.* **30**(4), 1231–1252 (2000)
18. Fomin, F.V., Gaspers, S., Lokshtanov, D., Saurabh, S.: Exact algorithms via monotone local search. In: *Proceedings of STOC 2016*, pp. 764–775 (2016)
19. Fomin, F.V., Golovach, P.A., Raymond, J.: On the tractability of optimization problems on h-graphs. In: *26th Annual European Symposium on Algorithms, ESA 2018, August 20–22, 2018, Helsinki, Finland*, pp. 30:1–30:14 (2018)
20. Fomin, F.V., Heggenes, P., Kratsch, D., Papadopoulos, C., Villanger, Y.: Enumerating minimal subset feedback vertex sets. *Algorithmica* **69**(1), 216–231 (2014)
21. Garg, N., Vazirani, V.V., Yannakakis, M.: Multitway cuts in node weighted graphs. *J. Algorithms* **50**(1), 49–61 (2004)
22. Golovach, P.A., Heggenes, P., Kanté, M.M., Kratsch, D., Sæther, S.H., Villanger, Y.: Output-polynomial enumeration on graphs of bounded (local) linear mim-width. *Algorithmica* **80**(2), 714–741 (2018)
23. Golovach, P.A., Heggenes, P., Kratsch, D., Saei, R.: Subset feedback vertex sets in chordal graphs. *J. Discrete Algorithms* **26**, 7–15 (2014)
24. Golombic, M.C., Rotics, U.: On the clique-width of some perfect graph classes. *Int. J. Found. Comput. Sci.* **11**(3), 423–443 (2000)
25. Hols, E.C., Kratsch, S.: A randomized polynomial kernel for subset feedback vertex set. *Theory Comput. Syst.* **62**, 54–65 (2018)
26. Jacob, H., Bellitto, T., Defrain, O., Pilipczuk, M.: Close Relatives (Of Feedback Vertex Set), Revisited. In: Golovach, P.A., Zehavi, M. (eds.) *16th International Symposium on Parameterized and Exact Computation (IPEC 2021). Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 214, pp. 21:1–21:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021)
27. Jaffke, L., Kwon, O., Strømme, T.J.F., Telle, J.A.: Mim-width III: graph powers and generalized distance domination problems. *Theor. Comput. Sci.* **796**, 216–236 (2019)
28. Jaffke, L., Kwon, O., Telle, J.A.: 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*, pp. 42:1–42:14 (2018)
29. Jaffke, L., Kwon, O., Telle, J.A.: Mim-width II. The feedback vertex set problem. *Algorithmica* **82**(1), 118–145 (2020)
30. Kawarabayashi, K., Kobayashi, Y.: Fixed-parameter tractability for the subset feedback set problem and the s-cycle packing problem. *J. Combin. Theory Ser. B* **102**(4), 1020–1034 (2012)
31. Kim, K.H.: *Boolean Matrix Theory and Applications*, vol. 70. Dekker, New York (1982)



32. Kratsch, D., Müller, H., Todinca, I.: Feedback vertex set on AT-free graphs. *Discrete Appl. Math.* **156**(10), 1936–1947 (2008)
33. Oum, S.-I.: Rank-width and vertex-minors. *J. Combin. Theory Ser. B* **95**(1), 79–100 (2005)
34. Oum, S.-I.: Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms*, 5(1):Art. 10, 20 (2009)
35. Oum, S.-I., Sæther, S.H., Vatshelle, M.: Faster algorithms for vertex partitioning problems parameterized by clique-width. *Theor. Comput. Sci.* **535**, 16–24 (2014)
36. Oum, S.-I., Seymour, P.: Approximating clique-width and branch-width. *J. Combin. Theory Ser. B* **96**(4), 514–528 (2006)
37. Papadopoulos, C., Tzimas, S.: Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs. *Discrete Appl. Math.* **258**, 204–221 (2019)
38. Papadopoulos, C., Tzimas, S.: Subset feedback vertex set on graphs of bounded independent set size. *Theor. Comput. Sci.* **814**, 177–188 (2020)
39. Robertson, N., Seymour, P.D.: Graph minors. x. Obstructions to tree-decomposition. *J. Comb. Theory Ser. B* **52**(2), 153–190 (1991)
40. Sæther, S.H., Vatshelle, M.: Hardness of computing width parameters based on branch decompositions over the vertex set. *Theor. Comput. Sci.* **615**, 120–125 (2016)
41. Vatshelle, M.: New width parameters of graphs. PhD thesis, University of Bergen, Bergen, Norway (2012)
42. Yamazaki, K.: Inapproximability of rank, clique, boolean, and maximum induced matching-widths under small set expansion hypothesis. *Algorithms* **11**(11), 173 (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.