Feedback vertex set on AT-free graphs

Dieter Kratsch

Université de Metz, Laboratoire d'Informatique Théorique et Appliquée, 57045 Metz Cedex 01, France. Email: kratsch@univ-metz.fr

Haiko Müller

School of Computing, University of Leeds, Leeds, LS2 9JT, United Kingdom. Email: hm@comp.leeds.ac.uk

Ioan Todinca

Laboratoire d'Informatique Fondamentale d'Orléans (LIFO), Université d'Orléans, BP 6759, 45067 Orléans Cedex 2, France. Email: Ioan.Todinca@univ-orleans.fr

Abstract

We present a polynomial time algorithm to compute a minimum (weight) feedback vertex set for AT-free graphs, and extending this approach we obtain a polynomial time algorithm for graphs of bounded asteroidal number.

Key words: feedback vertex set, asteroidal triple-free graph

1 Introduction

A *feedback vertex set* (fvs) of an undirected graph $G = (V, E)$ is a set W containing at least one vertex of each cycle of G. Hence $W \subseteq V$ is a feedback vertex set of $G = (V, E)$ if and only if $G - W$ is a forest. The FEEDBACK VERTEX SET (FVS) problem is, given a graph G and an integer k, whether G contains a feedback vertex set of size at most k.

The FVS problem is NP-complete [10]. It is also definable in a logical language called LinEMSOL($\tau_{1,L}$), thus the weighted versions is linear time solvable on graphs of bounded clique-width [8]. Hence the problem is linear-time solvable for graphs of bounded treewidth, cographs, distance-hereditary graphs, etc.

There is a lot of research on algorithms for FVS. The problem is fixed-parameter tractable [9]. There has been a series of papers on approximation algorithms on minimum (weight) FVS resulting in a 2-approximation algorithm for minimum weight FVS [1]. Clearly FVS had been and still is a benchmark covering problem. It is also important since FVS approximation algorithms are used as subroutines in approximation algorithms for other typically network analysis problems. See *e.g.* [11] where a PTAS for FVS on planar graphs is constructed as a by-product.

Concerning the complexity of FVS on special graph classes the following is known. FVS is NP-complete for bipartite and for planar graphs. A minimum (weight) fvs can be computed by a polynomial time algorithm for interval graphs ($O(n+m)$) [16], permutation graphs ($O(nm)$) [14], cocomparability graphs ($O(n^2m)$) [15] and convex bipartite graphs ($O(n^2m)$) [15]. Furthermore FVS is mentioned as polynomial time solvable for chordal graphs and circular-arc graphs in [17]. J. Spinrad points out in [17] that the complexity of FVS on AT-free graphs in unknown.

Three independent vertices form an *asteroidal triple* (AT for short) if any two of them are connected by a path avoiding the neighbourhood of the third vertex. Graphs without asteroidal triples are called *AT-free*. AT-free graphs contain cocomparability graphs, permutation graphs and interval graphs. In an important paper Corneil, Olariu and Stewart were the first to study structural properties of AT-free graphs [7]. In the last ten years various papers on the complexity of NP-complete problems for AT-free graphs have been published, see *e.g.* [4–6,13]. Inspecting their results one finds out that there are LinEMSOL($\tau_{1,L}$)-definable problems that remain NP-complete on AT-free graphs. CLIQUE is NP-complete on AT-free graphs [5] and MINIMUM DOMINATING CLIQUE remains NP-complete even on cocomparability graphs [3]. This provides another motivation to study FVS on AT-free graphs.

Our paper is organised as follows. In Section 2 we summarise the main ingredients of an approach developed in [5] to design polynomial time algorithms for problems on AT-free graphs, see also [4,6]. In Sections 3–5 we present a polynomial time algorithm for FVS on AT-free graphs, based on this general approach. In Section 6 we show how to extend the approach to graphs of bounded asteroidal number and we obtain a polynomial time algorithm for FVS.

2 Preliminaries

We use standard graph theory notation throughout. We denote by $N(u)$ the neighborhood of a vertex u, and by $N[u]$ the closed neighborhood $N(u) \cup \{u\}$.

By a connected component we mean a maximal connected induced subgraph of a graph. Thus we denote a connected component of a graph G with vertex set C by $G[C]$. Consider a vertex u of G and let $G[C_1]$, $G[C_2]$, …, $G[C_p]$ be the connected components of $G - N[u]$. We denote $\mathcal{C}_G(u) = \{C_1, C_2, \ldots, C_p\}$. For nonadjacent vertices u and v of a graph $G = (V, E)$ we denote by $C_G(u, v)$ the subset of V inducing the connected component of $G - N[v]$ containing u. As usual we omit the index if this does not cause confusion. Note that $w \in C(u, v)$ if and only if $u \in C(w, v)$. Using this notation, the vertices u, v and w form an AT if and only if $u \in C(v, w)$, $v \in C(w, u)$ and $w \in C(u, v)$.

2.1 Blocks and Intervals

Let $G = (V, E)$ be a connected graph and $v \in V$. The pair (v, C) is called a *block* of G if $G[C]$ is a connected component of $G - N[v]$. Removing a closed neighbourhood $N[v]$ for some vertex v of G decomposes $G - N[v]$ into connected components. Informally, the graph G is decomposed into blocks.

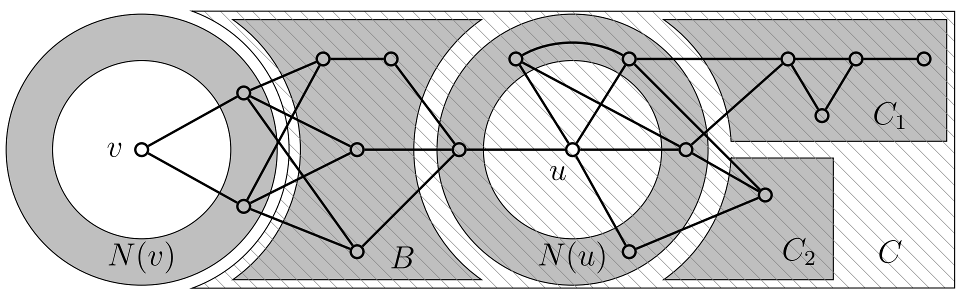

Fig. 1. Blocks and interval

Let u and v be two nonadjacent vertices of G and let $B = C(v, u) \cap C(u, v)$. The triple (v, B, u) is called an *interval* of G. Obviously (v, B, u) is an interval if and only if (u, B, v) is an interval. Notice that B might be empty.

Informally again, removing a closed neighbourhood from a block decomposes it into blocks and exactly one interval. Formally, consider a block (v, C) and let $u \in C$. Removing $N[u]$ from $G[C]$ partitions C into $B, C_1, \ldots, C_p$, where (v, B, u) is an interval and (u, C_i) are blocks of G, see Figure 1.

The FVS problem is solved by dynamic programming on blocks, following the technique developed in [5], see also [4,6]. We shall compute partial solutions for a block (v, C), *i.e.* solutions contained in $N[v] \cup C$. For this purpose we use partial solutions of blocks of type (u, C_i) with $u \in C$ and $C_i \subset C$. The vertex u will be chosen such that the partial solutions do not intersect the set B of the interval (u, B, v).

For solving the minimum feedback vertex set problem we shall actually compute the complement of this set, which is the largest induced forest. Hence we study induced forests $G[K]$ of G. Suppose that K is the set of vertices corresponding to a partial solution found by our algorithms and let v be a vertex of K. The neighbourhood of v in K is an independent set S of G. We would like to control $N(S) \setminus N[v]$, in order to understand how the forest $G[K]$ may be extended. In this subsection we show that of $N(S) \setminus N[v]$ can actually be identified by using at most four vertices of S; in particular polynomial space is sufficient for all possible combinations.

Definition 1. Consider a vertex v of the graph G and let S be an independent set contained in $N(v)$. We say that a set $R \subseteq S$ is a *representation of S w.r.t. v* if $N(S) \setminus N(v) = N(R) \setminus N(v)$ and R is of minimum size for this property. Similarly, if $G[C]$ is a component of $G - N[v]$, we say that R_C is a *representation of S in C* if $N(S) \cap C = N(R_C) \cap C$ and R_C is of minimum size for this property.

We continue with two easy observations. These will help us to prove that in AT-free graphs, each independent set contained in the neighbourhood of a vertex can be represented by at most four vertices.

Lemma 2. *Let $G = (V, E)$ be an AT-free graph and $v, y_1, y_2 \in V$ such that y_1 and y_2 belong to the same connected component of $G - N[v]$. If vertices $x_1 \in N(v) \cap N(y_1) \setminus N(y_2)$ and $x_2 \in N(v) \cap N(y_2) \setminus N(y_1)$ exist, then $y_1 y_2 \in E$.*

Proof. Suppose that $y_1 y_2 \notin E$, we claim that y_1, y_2 and v form an AT in G. Indeed, use the paths (v, x_1, y_1), (v, x_2, y_2) and a path from y_1 to y_2 in $G[C]$, each of these paths joins two of the three vertices and avoids the neighbourhood of the third. □

Lemma 3. *Let v be a vertex of an AT-free graph $G = (V, E)$ and let x_1, x_2, x_3 be three pairwise nonadjacent vertices in $N(v)$. Suppose that each x_i has a private neighbour y_i outside $N[v]$. That is, for $1 \leqslant i, j \leqslant 3$, $x_i y_j \in E$ if and only if $i = j$. Then y_1, y_2 and y_3 belong to exactly two different components of $G - N[v]$.*

Proof. Suppose first that y_1, y_2 and y_3 belong to the same component C of $G - N[v]$. By Lemma 2, y_1, y_2 and y_3 induce a triangle in G. Thus x_1, x_2 and x_3 form an AT of G—a contradiction.

Now assume that y_1, y_2, y_3 belong to three different components of $G - N[v]$. Clearly $\{y_1, y_2, y_3\}$ is an independent set and hence these vertices form an AT of G—a contradiction again. □

In the sequel we shall say that vertices x_1, x_2, x_3 and y_1, y_2, y_3 *violate Lemma 3* (w.r.t. vertex v) if $\{x_1, x_2, x_3\}$ is an independent subset of $N(v)$ such that for $i = 1, 2, 3$, $y_i \in V \setminus N[v]$ is a private neighbour of x_i w.r.t. $\{x_1, x_2, x_3\}$, and y_1, y_2, y_3 belong to three different components of $G - N[v]$.

The corollary follows directly from the previous lemma.

Corollary 4. *Let $G[C]$ be a connected component of $G - N[v]$ where $G = (V, E)$ is an AT-free graph, $v \in V$ and $C \subset V$. The neighbourhood of an independent set $S \subseteq N(v)$ in C can be represented by a subset of size at most two, i.e. there is a set $R_C \subseteq S$ with $|R_C| \leqslant 2$ and $N(R_C) \cap C = N(S) \cap C$. The outside neighbourhood of S can be represented by a subset of size at most four, i.e. there is a set $R \subseteq S$ with $|R| \leqslant 4$ and $N(R) \setminus N(v) = N(S) \setminus N(v)$.*

Proof. If R_C is a minimum size subset of S such that $N(R_C) \cap C = N(S) \cap C$, then each vertex x of R_C has a private neighbour y in C, *i.e.* a neighbour such that $N(y) \cap R_C = \{x\}$. Otherwise, $R_C \setminus \{x\}$ also represents the neighbourhood of S in C. By Lemma 3, R_C has at most two elements.

For similar reasons, if R represents the outside neighbourhood of S, each vertex of R has a private neighbour outside $N[v]$. By Lemma 3 the vertices of R have private neighbours in at most two components and at most two in each component, hence $|R| \leqslant 4$. □

3 Analysis

In this section we introduce the basic tools for computing large induced forests. First we define *local maps* which provide, for each vertex v a compact (polynomial) representation of the intersections between its neighborhood and all possible induced forests containing v. Based on this notion, we show how to glue forests representing partial solutions, in order to obtain larger forests.

3.1 Local Map

Define

$$\mathrm{fvs}(G) = \min\{|W| : G - W \text{ is a forest}\} \tag{1}$$

and let K be a set of vertices inducing a forest in the graph G (think of it as a partial solution for our problem). For a fixed vertex $v \in K$, denote by L the set of leaves of $G[K]$ contained in $N(v)$ and let $R' = K \cap N(v) \setminus L$. Note that in the forest $G[K]$ every vertex in L is adjacent to v only, and every vertex in R' is adjacent to v and at least one vertex outside $N[v]$. Let R represent L

w.r.t. v, see Definition 1. We call the triple (v, R, R') a *local map* of K. In the example of Figure 2, page 9, the forest K is represented by bold lines. The triple $(v, \{b, d\}, \{g\})$ is a local map of K.

The local maps will be an important tool for gluing forests, in order to obtain larger forests (partial solutions). The last statement of the next lemma is one of the main technical results for our algorithm. It implies that the number of all possible local maps of an AT-free graph is polynomially bounded.

Lemma 5. *Let (v, R, R') be a local map of $K \ni v$ inducing a forest $G[K]$ in a graph G that might as well contain ATs. The following conditions hold:*

(1) $R \cup R' \subseteq N(v)$;
(2) $R \cup R'$ is an independent set of G;
(3) for $x, x' \in R \cup R'$, $N(x) \setminus N(v) \subset N(x') \setminus N(v)$ implies $x \in R$ and $x' \in R'$;
(4) if G is AT-free then $|R \cup R'| \leqslant 4$.

Proof. We start with the easy proofs of the first three conditions. Then we shall prove the last condition.

Condition (1): follows directly from the definition of a local map.

Condition (2): If $G[R \cup R']$ contains an edge xy, then x, y and v would form a cycle in $G[K]$, thus $R \cup R'$ is an independent set of G.

Condition (3): Firstly notice that if $x, x' \in R$ and $N(x) \setminus N(v) \subset N(x') \setminus N(v)$ then $N(R \setminus \{x\}) \setminus N(v) = N(R) \setminus N(v)$, contradicting the fact that R is a representation of $(K \cap N(v)) \setminus R'$ w.r.t. v.

Now suppose that $x \in R'$ and $x' \in R \cup R'$. Then x has a neighbour $z \in K \setminus N[v]$, which is adjacent to x' too. That is, $(v, x, z\ x')$ is a cycle in $G[K]$—a contradiction, Condition (3) follows.

Condition (4): Corollary 4 implies $|R| \leqslant 4$. Since G is AT-free, we also have $|R'| \leqslant 2$. Indeed, if R' contains three vertices u_1, u_2, u_3, each u_i has a neighbour w_i in $K \setminus N[v]$ and w_1, w_2, w_3 form an asteroidal triple. Hence $|R \cup R'| \leqslant 6$ for AT-free graphs. Now let us show that actually $|R \cup R'| \leqslant 4$ holds for AT-free graphs. To do this we first prove three claims and then we shall prove that $|R \cup R'| > 4$ is impossible in AT-free graphs by excluding two cases.

Let $p = |R|$ and $q = |R'|$. We denote by $r_1, \ldots, r_p$ the vertices of R and by $r'_1, \ldots, r'_q$ the vertices of R'. For each $j = 1, 2, \ldots, q$, let $c'_j \in K \setminus N[v]$ be a vertex adjacent to r'_j. For each $i = 1, 2, \ldots, p$, let $c_i \in V \setminus N[v]$ be a vertex such that r_i is the only neighbour of c_i in R. By definition of a local map the vertices $c_1, c_2, \ldots c_p$ and $c'_1, c'_2, \ldots, c'_q$ exist. By construction $|\{c_1, c_2, \ldots, c_p\}| = p$.

Claim 1: For each $j = 1, 2, \ldots, q$, the only neighbour of c'_j in $R \cup R'$ is r'_j.

Proof. If c'_j had an additional neighbour $\hat{r} \in R \cup R'$, then $G[K]$ would contain a cycle $(c'_j, r'_j, v, \hat{r})$—a contradiction. $\Diamond$

Consequently, $\{c'_1, c'_2, \ldots, c'_q\}| = q$.

Claim 2: For each $j = 1, 2, \ldots q$, the vertex r'_j has at most one neighbour in $\{c_1, \ldots, c_p\}$.

Proof. Conversely, suppose that there is a $j \in \{1, 2, \ldots, q\}$ and $i_1, i_2 \in \{1, 2, \ldots, p\}$ such that $i_1 \neq i_2$ and r'_j is adjacent to c_{i_1} and c_{i_2}. Then r_{i_1}, r_{i_2} and c'_j form an AT which can be shown by the following paths: (r_{i_1}, v, r_{i_2}) avoiding $N[c'_j]$ since $c'_j \in V \setminus N[v]$ and by Claim 1; $(r_{i_1}, c_{i_1}, r'_j, c'_j)$ avoiding $N[r_{i_2}]$ by Condition (2) and the choice of c_{i_1}; and $(r_{i_2}, c_{i_2}, r'_j, c'_j)$ avoiding $N[r_{i_1}]$ by Condition (2) and the choice of c_{i_2}. $\Diamond$

Claim 3: For each component C of the graph $G - N[v]$ the following holds:

(1) $|C \cap \{c'_1, \ldots, c'_q\}| \leqslant 1$.
(2) $|C \cap \{c'_1, \ldots, c'_q\}| = 1$ implies $|C \cap \{c_1, \ldots, c_p\}| \leqslant 1$.

Proof. Contrary to (1) assume different vertices $c'_i, c'_j \in C$. According to Lemma 2, c'_i and c'_j are adjacent, thus $(c'_i, r'_i, v, r'_j, c'_j)$ is a cycle in $G[K]$—a contradiction.

Contrary to (2) assume three vertices $c_s, c_t, c'_j \in C$. We shall show that r_s, r_t and c'_j form an AT of G. Clearly (r_s, v, r_t) is a path in $G - N[c'_j]$ by Claim 1, *i.e.* (r_s, v, r_t) avoids $N[c'_j]$.

Next we construct a path from c'_j to r_{i_1} avoiding $N[r_{i_2}]$. If r'_j and c_s are adjacent, then the path (c'_j, r'_j, c_s, r_s) avoids $N[r_t]$. Otherwise c'_j and c_s are adjacent by Lemma 2. Thus the path (c'_j, c_s, r_s) avoids $N[r_t]$. By a symmetric argument there is a path from c'_j to r_{i_2} that avoids $N[r_{i_1}]$. That is, r_{i_1}, r_{i_2} and c'_j form an AT contradicting the fact that G is AT-free. $\Diamond$

Finally we are able to finish the proof of Condition (4): $p + q \leqslant 4$. Recall that $p \leqslant 4$ and $q \leqslant 2$. Suppose on the contrary that $p + q \geqslant 5$. We distinguish two cases, in the first case $p = 4$ and $q \geqslant 1$, in the second case $p \geqslant 3$ and $q = 2$.

Case 1: $p = 4$ and $q \geqslant 1$.

According to Lemma 3, the vertices c_1, c_2, c_3, c_4 belong to two different components of $G - N[v]$, and each of these components contains two of these vertices. We may assume $c_1, c_2 \in C_1$ and $c_3, c_4 \in C_2$. By Claim 3, c'_1 is in a third component of $G - N[v]$, say C_3. Since r'_1 is adjacent to at most one of the vertices in $\{c_1, c_2, c_3, c_4\}$ by Claim 2, we may suppose that r'_1 is adjacent neither to c_1, nor to c_3. This contradicts Lemma 3: The vertices r_1, r_3, r_1' and c_1, c_3, c_1' violate this lemma w.r.t. vertex v.

Case 2: $q = 2$ and $p \geqslant 3$.

As previously we may assume $c_1, c_2 \in C_1$, $c_1' \in C_2$ and $c_2' \in C_3$, where C_1, C_2 and C_3 are three different components of $G - N[v]$. Clearly $c_1', c_2' \notin E$ since both vertices belong to different components of $G - N[v]$. For the sake of a contradiction we shall show that c_1', c_2' and r_3 form an AT in G.

First note that the path (c_1', r_1', v, r_3) avoids $N[c_2']$ since $c_1', c_2' \notin E$ and by Claim 1. Similarly, the path (c_2', r_2', v, r_3) avoids $N[c_1']$ since $c_1', c_2' \notin E$ and by Claim 1. Finally we need to find a path from c_1' to c_2' avoiding $N[r_3]$. If c_1 is adjacent neither to r_1' nor to r_2', then the vertices r_1, r_1', r_2' and c_1, c_1', c_2' violate Lemma 3. (Note that c_1, c_1', c_2' belong to three different components of $G - N[v]$.) Similarly, if c_2 is adjacent neither to r_1' nor to r_2' then r_2, r_2', r_1' and c_2, c_2', c_1' violate Lemma 3. Hence c_1 is adjacent to either r_1' or r_2', and c_2 is adjacent to either r_1' or r_2'. Furthermore by Claim 2, neither r_1' nor r_2' is adjacent to both c_1 and c_2. Recall that $c_1c_2 \in E$ by Lemma 2. Consequently there is a path from c_1' to c_2' avoiding $N[r_3]$ whose vertices form a subset of $\{c_1', r_1', c_1, c_2, r_2', c_2'\}$.

Thus both cases lead to a contradiction in AT-free graphs which implies Condition (4). $\square$

3.2 Atlas

For a graph $G = (V, E)$ we consider triples (v, R, R') consisting of a vertex $v \in V$ and subsets $R, R' \subseteq N(v)$. The set of all such triples that satisfy the conditions of Lemma 5 is called the *atlas* of $G = (V, E)$ and denoted by $\mathcal{R}(G)$ or simply $\mathcal{R}$.

Definition 6. Let (v, R, R') be in the atlas of $G = (V, E)$. We say that $K \subseteq V$ is *compatible* with (v, R, R') if

- $G[K]$ is a forest and $L \subseteq K$ is the set of its leaves;
- $\{v\} \cup R \cup R' \subseteq K$;
- $N(v) \cap K \setminus L \subseteq R'$;
- R is a representation of $(K \cap N(v)) \setminus R'$, w.r.t. v.

Let $\mathcal{K}(v, R, R')$ denote the set of all sets K compatible with (v, R, R').

For each triple (v, R, R') in the atlas of G, the set $K_0 = \{v\} \cup R \cup R'$ induces a star in G. Therefore K_0 is compatible with (v, R, R'). All $K \in \mathcal{K}(v, R, R')$

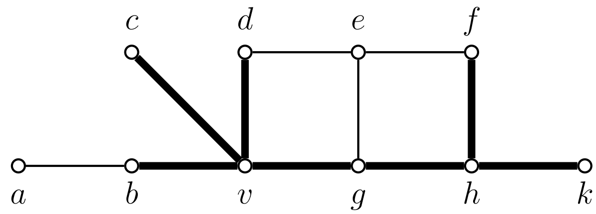

Fig. 2. Local maps in the atlas

contain K_0 as a subset. The only vertices of $K \cap N(v)$ which may have a neighbour in $K \setminus N[v]$ are the elements of R'. We emphasise that the vertices of R' are not required to have neighbours in $K \setminus N[v]$, they are just the only ones that *may* have neighbours in $K \setminus N[v]$. In Figure 2, the forest K is compatible with both $(v, \{b, d\}, \{g\})$ and $(v, \{d\}, \{b, g\})$. The former is a local map of the forest $K \cup \{a\}$. In other words, every induced forest containing the vertex v is compatible with its local map (v, R, R'), but some vertices in R' might be leaves of another forest compatible with (v, R, R').

Our algorithm will compute, for each triple (v, R, R') in the atlas and for each block (v, C), the size of a largest forest contained in $N[v] \cup C$ that is compatible with (v, R, R'). The following lemma shows in particular that it suffices to compute forests $K_1, K_2 \in \mathcal{K}(v, R, R')$ maximising the sizes of $K_1 \cap N(v)$ and $K_2 \cap C$.

As the remaining lemma in this subsection, it will be used in the forthcoming correctness proofs. As we decompose the input graph into blocks and intervals, we compose an induced forest in the graph from smaller ones in the components separated by closed neighbourhoods.

Informally, the next lemma suggests that for computing a maximum induced forest compatible with a local map (v, R, R'), we have to

- compute a forest compatible with (v, R, R') and having a maximum size intersection with $N(v)$;
- compute, for each block (v, C), a forest contained in $N[v] \cup C$, compatible with (v, R, R') and having a maximum size intersection with C;
- glue them together.

Lemma 7 (interchange). *Let $K_1, K_2 \subseteq V$ be compatible with $(v, R, R') \in \mathcal{R}(G)$. For every block (v, C) of G, $K_3 = (K_1 \setminus C) \cup (K_2 \cap C)$ is compatible with (v, R, R').*

Proof. We only show that K_3 induces a forest in G because the other conditions are obviously fulfilled. By contradiction let $Z \subseteq K_3$ induce a shortest cycle in $G[K_3]$. Clearly $Z \cap C \neq \varnothing$ (otherwise $Z \subseteq K_1$) and $Z \cap N[v] \neq \varnothing$ (otherwise $Z \subseteq K_2$). Let $(a, x_1, \ldots, x_p, b)$ be a subpath of Z such that $x_i \in C$ for all i with $1 \leqslant i \leqslant p$, and $a, b \in N(v)$ (possibly $a = b$). Since $a, b \in K_1 \backslash C$, there are vertices $a', b' \in R \cup R'$, such that $x_1 \in N(a')$ and $x_p \in N(b')$. If $a' = b'$, then $(a', x_1, \ldots, x_p)$ is a cycle of G, else $(v, a', x_1, \ldots, x_p, b')$ is a cycle of G, contradicting the fact that $v, a', b', x_1, \ldots, x_p \in K_2$ and $G[K_2]$ is acyclic. $\square$

Suppose we are given a forest K_v compatible with a local map (v, R, R') and let us fix a component $G[C]$ of $G - N[v]$. One of our main tasks will be to extend K_v in order to make its intersection with C as large as possible. Moreover, assume that we have previously computed, for some vertex $w \in C$, a forest K_w compatible with (v, R, R') having a large intersection with C. According to Lemma 8, these forests can be glued together (respecting some technical condition). We will later use this technique to compute a forest compatible with (v, R, R') and having a maximum intersection with C (see Lemma 12).

Lemma 8 (extension)**.** *Let v and w be two non-adjacent vertices of $G = (V, E)$ and let $K_v, K_w \subseteq V$ be compatible with $(v, R, R'), (w, Q, Q') \in \mathcal{R}(G)$, respectively. Assume that $K_0 = \{v, w\} \cup R \cup R' \cup Q \cup Q'$ induces a forest compatible with both (v, R, R') and (w, Q, Q'). Then*

(1) $K_v \cap N(v) \cap N(w) = K_w \cap N(w) \cap N(v) = R' \cap Q'$, and
(2) $K = (K_v \setminus C(w, v)) \cup (K_w \setminus C(v, w))$ is compatible with both (v, R, R') and (w, Q, Q').

Proof. We assume $K_0 \in \mathcal{K}$ where $\mathcal{K} = \mathcal{K}(v, R, R') \cap \mathcal{K}(w, Q, Q')$. To prove (1) we show that $K_v \cap N(v) \cap N(w) = R' \cap Q'$. Clearly $R' \cap Q' \subseteq K_v \cap N(v) \cap N(w)$. Conversely, let $u \in K_v \cap N(v) \cap N(w)$. In the forest $G[K_0]$, u is adjacent to both v and w, so it does not correspond to a leaf. Since K_0 is compatible with (v, R, R'), we have $u \in R'$. Since K_0 is compatible with (w, Q, Q') too, we deduce $u \in Q'$. Therefore $K_v \cap N(v) \cap N(w) = R' \cap Q'$, and by symmetry also $K_w \cap N(w) \cap N(v) = R' \cap Q'$.

To prove (2) we apply Lemma 7 to the block $(v, C(w, v))$ with $K_1 = K_v$, $K_2 = K_0$ and $K_3 = K'_v$. We obtain that $K'_v = (K_v \backslash C(w, v)) \cup K_0 \in \mathcal{K}(v, R, R')$. Note that $K'_v \cap N[w] = K_0 \cap N[w] = \{w\} \cup Q \cup Q'$. Moreover, the only vertices of $K'_v \cap N(w)$ which are not leaves in $G[K_v]$ are adjacent to both v and w, hence they are in Q'. So we have $K'_v \in \mathcal{K}(w, Q, Q')$. For similar reasons, $K'_w = (K_w \setminus C(v, w)) \cup K_0 \in \mathcal{K}$.

Finally we apply Lemma 7 to the block $(v, C(w, v))$, with $K_1 = K'_v$ and $K_2 = K'_w$ and obtain that $K \in \mathcal{K}(v, R, R')$. By Lemma 7 again, applied to $(w, C(v, w))$ with $K_1 = K'_w$ and $K_2 = K'_v$, we conclude that $K \in \mathcal{K}(w, Q, Q')$. $\square$

4 Synthesis

The gluing techniques based on local maps will be transformed in this section into explicit equations allowing to compute maximum induced forests. Let (v, C) be a block of an AT-free graph $G = (V, E)$ and $(v, R, R') \in \mathcal{R}(G)$. We define two quantities:

- $\mathsf{start}(v, R, R')$ is the maximum size of a subset of $N[v]$ compatible with (v, R, R').
- $\mathsf{branch}(v, R, R', C)$ is the maximum size of a subset of C that has a superset compatible with (v, R, R').

These quantities are sufficient to compute $\mathrm{fvs}(G)$.

More formally, we have:

$$\mathsf{start}(v, R, R') = \max_{K \in \mathcal{K}(v,R,R')} |K \cap N[v]| \,, \tag{2}$$

$$\mathsf{branch}(v, R, R', C) = \max_{K \in \mathcal{K}(v,R,R')} |K \cap C| \,. \tag{3}$$

Recall that $\mathcal{C}(v)$ denotes the set of vertex sets of the connected components of $G - N[v]$.

Lemma 9.

$$\mathrm{fvs}(G) = |V| - \max_{(v,R,R') \in \mathcal{R}(G)} \Big(\mathsf{start}(v, R, R') + \sum_{C \in \mathcal{C}(v)} \mathsf{branch}(v, R, R', C) \Big) \tag{4}$$

Proof. Let K be the vertex set of a largest induced forest of G and let v be any vertex in K. By Lemma 5 the local map (v, R, R') of K belongs to the atlas $\mathcal{R}(G)$. By Equation (1), it is sufficient to prove that $|K|$ equals the "max" part in the equation above. Directly from Equations (2) and (3) follows that $|K| = |K \cap N[v]| + \sum_{C \in \mathcal{C}(v)} |K \cap C| \leqslant \mathsf{start}(v, R, R') + \sum_{C \in \mathcal{C}(v)} \mathsf{branch}(v, R, R', C)$.

Conversely, let $K_v \in \mathcal{K}(v, R, R')$ such that $|K_v \cap N[v]| = \mathsf{start}(v, R, R')$, and for each $C \in \mathcal{C}(v)$ let $K_C \in \mathcal{K}(v, R, R')$ such that $|K_C \cap C| = \mathsf{branch}(v, R, R', C)$. By Lemma 7, $(K_v \cap N[v]) \cup \Big(\bigcup_{C \in \mathcal{C}(v)} (K_C \cap C) \Big)$ induces a forest compatible with (v, R, R') and the conclusion follows. $\square$

We shall prove that the quantities start can be computed independently for all triples in the atlas. The value of $\mathsf{branch}(v, R, R', C)$ can be computed recursively using values of the type $\mathsf{branch}(w, Q, Q', D)$ for blocks (w, D) such that $D \subset C$. Eventually we compute $\mathrm{fvs}(G)$ using Equation (4).

4.1 *Computing* start

As usual $\alpha(G)$ denotes the maximum size of an independent set $S \subseteq V$ of $G = (V, E)$. For a triple (v, R, R') in the atlas of G we define

$$\mathsf{Inf}(v, R, R') = \{x \in N(v) : N(x) \setminus N(v) \subseteq N(R)\} \setminus N[R \cup R'] \,. \tag{5}$$

Lemma 10 (start)**.**

$$\mathsf{start}(v, R, R') = 1 + |R \cup R'| + \alpha(G[\mathsf{Inf}(v, R, R')]) \tag{6}$$

holds for all $(v, R, R') \in \mathcal{R}(G)$.

Proof. Let rhs_6 be shorthand for the term on the right hand side of Equation (6). Firstly we shall show $\mathsf{start}(v, R, R') \geqslant \mathrm{rhs}_6$. Therefore let S be a maximum independent set of $G[\mathsf{Inf}(v, R, R')]$. Then $R \cup R'$ is an independent set by Condition (2) of Lemma 5, and $R \cup R' \cup S$ is an independent set the definition of Inf in Equation (5). Hence the sets $\{v\}$, R, R' and S are pairwise disjoint and their union belongs to $\mathcal{K}(v, R, R')$. We conclude $\mathsf{start}(v, R, R') \geqslant \mathrm{rhs}_6$.

On the other hand consider any $K \in \mathcal{K}(v, R, R')$. We have $\{v\} \cup R \cup R' \subseteq K$ by Definition 6. Moreover $|K \cap N[v] \setminus (\{v\} \cup R \cup R')| \leqslant |S|$ holds by Definition 6 again and our choice of the set S. This implies $\mathsf{start}(v, R, R') \leqslant \mathrm{rhs}_6$. ☐

4.2 *Computing* branch

Lemma 11. *Let $G = (V, E)$ be an AT-free graph and $K \subseteq V$. Consider a vertex $v \in K$ and a block (v, C) of G such that $K \cap C \neq \varnothing$. Then there is a vertex $w \in K \cap C$ such that $K \cap B = \varnothing$ for the interval (v, B, w).*

Proof. Let w be any vertex in $K \cap C$ and assume a vertex $u \in K \cap B$. We consider the interval (v, A, u). By an inductive argument it suffices to show $A \subseteq B$ because we have $u \in B \setminus A$.

By contradiction, let $z \in A \setminus B$. Since $u \in B$, a path $(v, \ldots, u)$ exists in $G - N[w]$ and a path $(w, \ldots, u)$ exists in $G - N[v]$. Moreover there is a path $(v, \ldots, z)$ in $G - N[u]$ because $z \in A$. The latter path meets $N(w)$ because $z \notin B$. That is, we have a path $(v, \ldots, w)$ in $G - N[u]$ too. Now, since u, v and w do not form an AT, such a vertex z cannot exist. ☐

The following Lemma allows us to compute the quantity $\mathsf{branch}(v, R, R', C)$ for a block (v, C) and a triple (v, R, R'), using the values $\mathsf{branch}(w, Q, Q', D)$ for smaller blocks (w, D).

Lemma 12 (branch). *For every block (v, C) of an AT-free graph G and all $(v, R, R') \in \mathcal{R}(G)$ we have*

$$\mathsf{branch}(v, R, R', C) = \max_{\substack{w \in C \\ (w,Q,Q') \in \mathcal{R}(G) \\ K_0 \in \mathcal{K}(v,R,R') \cap \mathcal{K}(w,Q,Q')}} \Big(\mathsf{start}(w, Q, Q') - |Q' \cap R'| + \\ + \sum_{D \in \mathcal{C}(w),\, D \subset C} \mathsf{branch}(w, Q, Q', D)\Big) \quad (7)$$

where $K_0 = \{v, w\} \cup R \cup R' \cup Q \cup Q'$. If there is no vertex $w \in C$ such that $Q, Q' \subseteq N(w)$ with $(w, Q, Q') \in \mathcal{R}(G)$ and $K_0 \in \mathcal{K}(v, R, R') \cap \mathcal{K}(w, Q, Q')$ exist, we have $\mathsf{branch}(v, R, R', C) = 0$.

Proof. We use rhs_7 to abbreviate the term on the right hand side of Equation (7). Firstly we will show $|K \cap C| \leqslant \mathrm{rhs}_7$ for a set K compatible with (v, R, R'). If $K \cap C \neq \varnothing$, by Lemma 11 there is a vertex $w \in C$ such that the interval between v and w does not contain vertices in K. Let (w, Q, Q') be the corresponding local map of K. For each component $G[D]$ of $G - N[w]$ we have $|K \cap D| \leqslant \mathsf{branch}(w, Q, Q', D)$ by Equation (3). Summing these values up for all $D \not\ni v$ (or equivalently, for all $D \subset C$) we obtain $|K \cap \bigcup_{D \subset C} D| \leqslant \sum_D \mathsf{branch}(w, Q, Q', D)$.

Note that $K \cap N(v) \cap N(w) = R' \cap Q'$, by Lemma 8 (1) applied to $K_v = K_w = K$. Now Equation (2) implies $|K \cap N[w] \setminus N[v]| \leqslant \mathsf{start}(w, Q, Q') - |Q' \cap R'|$. Since the interval between v and w does not contain vertices in K, it follows $|K \cap C| \leqslant \mathrm{rhs}_7$.

For the other way around, let the maximum in Equation (7) be attained for $w \in C$ and the triple $(w, Q, Q') \in \mathcal{R}(G)$. For each block (w, D) with $D \subset C$

- let $K_D \in \mathcal{K}(w, Q, Q')$ be a subset of D with $|K_D \setminus N[w]| = \mathsf{branch}(w, Q, Q', D)$,
- let $K_w \in \mathcal{K}(w, Q, Q')$ be a subset of $N[w]$ with $|K_w| = \mathsf{start}(w, Q, Q')$, and
- let $K_v \in \mathcal{K}(v, R, R')$ be a subset of $N[v]$ such that $|K_v| = \mathsf{start}(v, R, R')$.

K_0 is compatible with (v, R, R') and (w, Q, Q'). Hence by Lemma 8 (2), the set $K_v \cup K_w$ is also compatible with both local maps. By Lemma 7, $K = K_v \cup K_w \cup \bigcup_{D \subset C} K_D$ is compatible with (w, Q, Q') and consequently also with (v, R, R'). By Lemma 8 (1), $K_w \cap N(v) = R' \cap Q'$, hence $|K_w \cap C| = \mathsf{start}(w, Q, Q') - |R' \cap Q'|$. We have that $|K \cap C| = |K_w \cup \bigcup_{D \subset C} K_D| \geqslant \mathrm{rhs}_7$. □

5 Algorithm

Our algorithm computes $\mathrm{fvs}(G)$ for AT-free graphs G using Equations (6) for computing start, (7) for branch and eventually (4) for $\mathrm{fvs}(G)$. The crucial observation is that in step 3, when computing $\mathsf{branch}(v, R, R', C)$ for a block (v, C), we only need the values of $\mathsf{branch}(w, Q, Q', D)$ for blocks (w, D) such that $D \subset C$. Since the blocks (v, C) are sorted (in step 2) by $|C|$, the block (w, D) appears before (v, C) in step 3.

Algorithm `FeedbackVertexSet`
Input: an AT-free graph G.
Output: $\mathrm{fvs}(G)$.

step 1 For all $(v, R, R') \in \mathcal{R}(G)$ compute $\mathsf{start}(v, R, R')$ using Equation (6).

step 2 Create a list of all blocks (v, C) of G and sort by $|C|$.

step 3 For each block (v, C) of G and each triple $(v, R, R') \in \mathcal{R}(G)$ compute $\mathsf{branch}(v, R, R', C)$ according to Equation (7) using values $\mathsf{start}(w, Q, Q')$ computed in step 1 and values $\mathsf{branch}(w, Q, Q', D)$ computed earlier in this step.

step 4 Determine $\mathrm{fvs}(G)$ by Equation (4) using values $\mathsf{branch}(v, R, R', C)$ computed in step 3.

Theorem 13. *There is an $\mathcal{O}(n^8 m^2)$ algorithm to compute the minimum size of a feedback vertex set in AT-free graphs.*

Proof. The correctness of this algorithm follows from Lemmas 10 and 12. We analyse the running time for an input $G = (V, E)$ where $n = |V|$, $m = |E|$ and n_v is the degree $d_G(v)$ of $v \in V$.

step 1 For each $(v, R, R') \in \mathcal{R}$ we have $|R \cup R'| \leqslant 4$, see Condition (4) in Lemma 5. Hence we have $\mathcal{O}(n_v^4)$ triples for each $v \in V$. This adds up to $|\mathcal{R}| = \mathcal{O}(n^3 m)$ because $\sum_{v \in V} n_v = 2m$.

We compute $\alpha(G[\mathsf{Inf}(v, R, R')])$ in time $\mathcal{O}(n_v^3)$ time because G is AT-free [13]. So the total time for this step is $\mathcal{O}(n^6 m)$.

step 2 The number of blocks is $\mathcal{O}(n^2)$ and for fixed v we find all blocks (v, C) in time $\mathcal{O}(n + m)$. Hence this step needs time $\mathcal{O}(nm + n^2)$.

step 3 Here we compute $\mathcal{O}(n^4 m)$ values $\mathsf{branch}(v, R, R', C)$. For each one we minimise over $\mathcal{O}(n^3 m)$ tiples (w, Q, Q'). Since $|Q \cup Q'| \leqslant 4$ we look up a single value $\mathsf{start}(w, Q, Q')$ or $\mathsf{branch}(w, Q, Q', D)$ in constant time. Consequently this step takes $\mathcal{O}(n^8 m^2)$ time.

step 4 The last step is similar to step 3 and takes time $\mathcal{O}(n^7 m^2)$.

Hence the overall running time can be bounded by $\mathcal{O}(n^8 m^2)$. ☐

The following definition from [12] generalises asteroidal triples.

Definition 14. An independent set A of G is called *asteroidal set* if for every vertex $a \in A$ there is a connected component $G[C]$ of $G - N[a]$ such that $A \setminus \{a\} \subseteq C$. The *asteroidal number* of G, denoted by $\mathrm{an}(G)$, is the maximum cardinality of an asteroidal set of G.

It is easy to see that an independent set is asteroidal if and only if each three-element subset forms an AT. The following Ramsey-type lemma will be used in the proof of Lemma 16.

Lemma 15. *Each graph on more than $2k^2$ vertices has $k+1$ pairwise nonadjacent vertices or a 2-connected subgraph on $k+1$ vertices.*

Proof. Let $G = (V, E)$ be any graph on $|V| > 2k^2$ vertices that has no 2-connected subgraph on $k+1$ vertices. By $\mathcal{B}$ we denote the set of all subsets $B \subseteq V$ such that $G[B]$ is a maximal subgraph of G without cut vertex. Two different sets $A, B \in \mathcal{B}$ are either disjoint or their intersection contains just one cut vertex of G. Clearly $|\mathcal{B}| \geqslant \lceil (2k^2+1)/k \rceil = 2k+1$. The set of cut vertices of G is denoted by C. We construct the bipartite graph $T = (\mathcal{B}, C, F)$ with $\{B, c\} \in F$ if $c \in B$ which is is fact a forest [18].

We partition $\mathcal{B}$ into *layers*. From each connected component of T we choose a subset of V in $\mathcal{B}$. The layer L_0 is the set of these subsets. Define $L_i = \{B : d_T(B_0, B) = 2i \text{ for some } B_0 \in L_0\}$, $i \geqslant 0$. Clearly the layers of odd parity or the levels of even parity contain at least $k+1$ elements. Suppose w.l.o.g. that we are in the latter case. For each element $A \in \mathcal{B}$, let $c(A)$ be the cut vertex next to A on the path from A to the level 0 in the forest T. ($c(A)$ does not exist if $A \in L_0$). Also fix a vertex $x(A) \in A \setminus \{c(A)\}$. We show that the set $\{x(A) \mid A \in L_{2j}\}$ is an independent set of G, of size at least $k+1$. Let $A \in L_{2j}, B \in L_{2j'}$, $0 \leqslant 2j \leqslant 2j'$. We prove that $x(A)$ and $x(B)$ are distinct and non adjacent in G. Only consider the case when A and B are in a same component of T. If A is on the unique path of T from B to an element of L_0, observe that $c(B)$ separates $x(A)$ and $x(B)$ in G: indeed, $c(B)$ separates the nodes A and B of T, and since A is not adjacent to $c(B)$ in T, we have $x(A) \neq c(B)$ and the conclusion follows. If $A \cap B \neq \varnothing$ the common cut vertex separates $x(A)$ and $x(B)$. Otherwise consider $c(C)$ for the lowest common ancestor C of A and B in T. By construction $c(C)$ is different from $x(A)$ and $x(B)$ and c separates these two vertices in G. □

Recall that, by Definition 1, the *representation* of an independent set $S \subseteq N(v)$ is a minimum size set $R \subseteq S$ such that $N(S) \setminus N(v) = N(R) \setminus N(v)$. The following lemma generalises Corollary 4.

Lemma 16. *For all vertices v of graph G with $\mathrm{an}(G) \leqslant k$, every independent set $S \subseteq N(v)$ has a representation of size at most $2k^2$.*

Proof. Conversely assume that S is an independent set containing more than $2k^2$ vertices such that S is its own representation. Then every vertex $s \in S$ has a private neighbour $t \in N(s) \setminus (N(v) \cup \bigcup_{s' \in S \setminus \{s\}} N(s'))$. Let T be the set of these private neighbours. Since $|T| = |S| > 2k^2$, Lemma 15 ensures that $G[T]$ contains an independent set T' with $|T'| = k+1$ or a 2-connected subgraph $G[C]$ on $k+1$ vertices. In the latter case let $S' \subseteq S$ be the set of private neighbours of C. In both cases T' and S', respectively, form an asteroidal set of G contradicting $\mathrm{an}(G) \leqslant k$. ☐

Here we generalise our approach in Section 3 to graphs of bounded asteroidal number. First we reconsider Lemma 5. Let $G[K]$ be a forest in G containing the vertex v. We consider the local map (v, R, R') of K, *i.e.* R' is the set of internal vertices of $G[K]$ belonging to $N(v)$ and R is a representation of $(K \cap N(v)) \setminus R'$.

Lemma 17. *Let (v, R, R') be a local map of $K \ni v$ inducing a forest $G[K]$ in G. The following conditions hold:*

(1) $R \cup R' \subseteq N(v)$;
(2) $R \cup R'$ is an independent set of G;
(3) $N(x) \setminus N(v) \subset N(x') \setminus N(v)$ implies $x \in R$ and $x' \in R'$ for $x, x' \in R \cup R'$;
(4) $|R| \leqslant 2k^2$ and $|R'| \leqslant k$ where $k = \mathrm{an}(G)$.

Proof. Conditions (1)–(3) are the same as in Lemma 5. $|R| \leqslant 2k^2$ follows from Lemma 16. It remains to prove $|R'| \leqslant k$. For each $s \in R'$ we choose a neighbour t in $K \setminus N[v]$. Let T be the set of these neighbours. Then $G[R' \cup T]$ contains exactly $|R'|$ edges, *i.e.* T is an independent set of private neighbours. In other words, T is an asteroidal set of G. Condition (4) follows. ☐

Now the *atlas* of G is the set of all triples (v, R, R') satisfying the conditions of Lemma 17. It is denoted by $\mathcal{R}(G)$ again. We retain the concept of *compatibility* as introduced in Definition 6.

Lemma 7 holds for all graphs. We extend Lemma 8 (2) as follows. For an asteroidal set A of G consider the set $B(A) = \bigcap_{a,b \in A, a \neq b} C(b, a)$. We say that $(A, B(A))$ is an *interval* of G. Especially every block (v, C) is an interval $(\{v\}, C)$.

Lemma 18. *Let (A, B) be an interval of $G = (V, E)$ and for all $a \in A$ let $K_a \subseteq V$ be compatible with $(a, R_a, R'_a) \in \mathcal{R}(G)$. Then $K = \bigcup_{a\in A}(K_a \setminus \bigcup_{b\in A\setminus\{a\}} C(b, a))$ is compatible with (a, R_a, R'_a) for all $a \in A$ if $K_0 = A \cup \bigcup_{a\in A}(R_a \cup R'_a)$ is compatible with (a, R_a, R'_a) for all $a \in A$.*

Proof. As in the proof of Lemma 7 it suffices to show that $G[K]$ is a forest. Conversely, let $Z \subseteq K$ induce a shortest cycle in $G[K]$. Then Z meets $N[v]$. □

We keep the definition of start and branch (Equations (2) and (3)). Since Lemma 10 holds for all graphs we can compute $\mathsf{start}(v, R, R')$ as before, see Equation (6). To compute $\mathsf{branch}(v, R, R', C)$ we need the following generalisation of Lemma 11.

Lemma 19. *Let $K \subseteq V$ be any set of vertices in a graph $G = (V, E)$, $v \in K$ and $(\{v\}, C)$ an interval of G. Then there is a set $A' \subseteq K \cap C$ such that $A' \cup \{v\}$ is asteroidal and $K \cap C \cap \bigcap_{a\in A'} C(v, a) = \varnothing$.*

Proof. Let $A' \subseteq K \cap C$ be any set such that $A = A' \cup \{v\}$ is asteroidal, for instance, $A' = \varnothing$. Let $\mathsf{Int}(A) = C \cap \bigcap_{a\in A'} C(v, a)$. For a vertex $u \in \mathsf{Int}(A)$ we define $A \oplus u = (A \cap C(v, u)) \cup \{u\}$. Obviously $A \oplus u$ is an asteroidal set of $G[N[v] \cup C]$ containing the vertices u and v. As in the proof of Lemma 11 we apply an inductive argument. It suffices to show $\mathsf{Int}(A \oplus u) \subseteq \mathsf{Int}(A)$ for a vertex $u \in K \cap \mathsf{Int}(A)$ because $u \notin \mathsf{Int}(A \oplus u)$.

To prove $\mathsf{Int}(A\oplus u) \subseteq \mathsf{Int}(A)$ we show $C(v, u) \subseteq C(v, a)$ for all $a \in A\setminus(A\oplus u)$. Let $a \in A$ be such a vertex, *i.e.* v and a belong to different connected components of $G - N[u]$. Since $u \in \mathsf{Int}(A)$ we have $u \in C(v, a)$, and consequently $C(v, u) \subseteq C(v, a)$. □

The following lemma allows us to compute branch in a recursive manner.

Lemma 20. *For every block (v, C) of a graph G and all $(v, R, R') \in \mathcal{R}(G)$ we have*

$$\mathsf{branch}(v, R, R', C) = \max_{\substack{A'\subseteq C \\ A \text{ is asteroidal} \\ A\cup \bigcup_{a\in A}(R_a\cup R'_a)\in \bigcap_{a\in A}\mathcal{K}(a,R_a,R'_a)}} \Big(|\bigcup_{a\in A'} R'_a \setminus R'| + \sum_{a\in A'} \Big(\mathsf{start}(a, R_a, R'_a) - |R'_a| + \sum_{D\in\mathcal{C}(a),\, D\subset C} \mathsf{branch}(a, R_a, R'_a, D)\Big)\Big) \quad (8)$$

where $A = A' \cup \{v\}$, $R_v = R$, $R'_v = R'$ and $(a, R_a, R'_a) \in \mathcal{R}(G)$ for all $a \in A'$.

Proof. Let K induce a forest in $G[C \cup N[v]]$ that is compatible with (v, R, R'). By Lemma 19 there is a set $A' \subseteq K \cap C$ such that $A = A' \cup \{v\}$ is asteroidal and $K \cap \mathsf{Int}(A) = \varnothing$. For each $a \in A'$ let (a, R_a, R'_a) be the corresponding local map of K. Then for each connected component $G[D]$ of $G - N[a]$ with $v \notin D$ (or, equivalently, with $D \subset C$) we have $|K \cap D| \leqslant \mathsf{branch}(a, R_a, R'_a, D)$. Summing up these values we obtain $|K \cap \bigcup_{D \not\ni v}| \leqslant \sum_{D \subset C} \mathsf{branch}(a, R_a, R'_a, D)$. We obtain the upper bound for $\mathsf{branch}(v, R, R', C)$ by summing over all $a \in A'$, whereby we include each vertex in K exactly once. Remember that different sets $N[a] \cup D$ overlap only at $N[a]$, and only vertices in R'_a have neighbours in $G[K]$ that do not belong to $N[a]$.

Now we consider set $A' \subseteq C$ such that $A = A' \cup \{v\}$ is asteroidal. For each $a \in A'$ and each $(a, R_a, R'_a) \in \mathcal{R}(G)$ let K_a induce a forest in $G - C(v, a)$ compatible with (a, R_a, R'_a) such that $|K_a| = \sum_{D \in \mathcal{C}(a), D \subset C} \mathsf{branch}(a, R_a, R'_a, D)$. If we have $A \cup \bigcup_{a \in A}(R_a \cup R'_a) \in \bigcap_{a \in A} \mathcal{K}(a, R_a, R'_a)$ then $K = \bigcup_{a \in A} K_a$ induces a forest in $G[N[v] \cup C]$ for each $K_v \subseteq N[v]$ compatible with (v, R, R'). This proves the lower bound because the size of K is bounded by the term on the right hand side of Equation 8. $\square$

From Lemmas 10 and 20 we derive an algorithm similar to the one given in Section 5. We claim that it runs in polynomial time whenever the asteroidal number of the input is bounded by a constant. The tricky point is to bound the size of $\mathcal{R}(G)$. Here Lemma 17 can help. Due to its Condition (4) we know $|R| \leqslant 2k^2$ and $|R'| \leqslant k$ for every triple $(v, R, R') \in \mathcal{R}(G)$ if $\mathrm{an}(G) \leqslant k$. Hence $|\mathcal{R}(G)| < n^{2k^2+k+1}$ where $k = \mathrm{an}(G)$.

References

[1] V. Bafna, P. Berman, T. Fujito: A 2-approximation algorithm for the undirected feedback vertex set problem, *SIAM Journal on Discrete Mathematics* **12** (1999), pp. 289–297.

[2] H. Bodlaender, D. Thilikos: Treewidth for graphs with small chordality. *Discrete Applied Mathematics* **79** (1997), pp. 45–61.

[3] A. Brandstädt, D. Kratsch: On domination problems for permutation and other graphs, *Theoretical Computer Science* **54** (1987), pp. 181–198.

[4] H.J. Broersma, A. Huck, T. Kloks, O. Koppius, D. Kratsch, H. Müller and H. Tuinstra: Degree-preserving trees, *Networks* **35** (2000), pp. 26–39.

[5] H.J. Broersma, T. Kloks, D. Kratsch and H. Müller: Independent sets in asteroidal triple-free graphs, *SIAM Journal on Discrete Mathematics* **12** (1999), pp. 276–287.

[6] H.J. Broersma, T. Kloks, D. Kratsch and H. Müller: A generalization of AT-free graphs and a generic algorithm for solving triangulation problems, *Algorithmica* **32** (2002), pp. 594–610.

[7] D.G. Corneil, S. Olariu and L. Stewart: Asteroidal triple-free graphs, *SIAM Journal on Discrete Mathematics* **10** (1997), pp. 399–430.

[8] B. Courcelle, J.A. Makowsky, U. Rotics: Linear time solvable optimization problems on graphs of bounded clique-width, *Theory of Computing Systems* **33** (2000), pp. 125–150.

[9] R.G. Downey, M.R. Fellows: *Parameterized complexity*, Springer, 1997.

[10] M.R. Garey and D.S. Johnson: *Computers and Intractability: A guide to the Theory of NP-completeness*, Freeman, New York, 1979.

[11] J. Kleinberg, A. Kumar: Wavelength conversion in optical networks, *Journal of Algorithms* **38** (2001) pp. 25–50.

[12] T. Kloks, D. Kratsch and H. Müller: On the structure of graphs with bounded asteroidal number, *Graphs and Combinatorics* **17** (2001), pp. 295–306.

[13] E. Köhler: Graphs without asteroidal triples, PhD thesis, TU Berlin 1999. `ftp://ftp.math.tu-berlin.de/pub/combi/ekoehler/diss`

[14] D.Y. Liang: On the feedback vertex set problem in permutation graphs, *Information Processing Letters* **52** (1994) pp. 123–129.

[15] D.Y. Liang, M.S. Chang: Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs, *Acta Informatica* **34** (1997) pp. 337–346.

[16] C.L. Lu, C.Y. Tang: A linear-time algorithm for the weighted feedback vertex problem on interval graphs, *Information Processing Letters* **61** (1997) pp. 107–111.

[17] J. Spinrad: *Efficient graph representations*, American Mathematical Society, Fields Institute Monograph Series 19, 2003.

[18] R. Tarjan: Depth-first search and linear graph algorithms. *SIAM Journal on Computing* **1** (1972) pp. 146–160.