

# Independent Feedback Vertex Sets for Graphs of Bounded Diameter\*

Marthe Bonamy<sup>1</sup>, Konrad K. Dabrowski<sup>2</sup>, Carl Feghali<sup>3</sup>,  
Matthew Johnson<sup>2</sup>, and Daniël Paulusma<sup>2</sup>

<sup>1</sup> CNRS, LaBRI, France [marthe.bonamy@u-bordeaux.fr](mailto:marthe.bonamy@u-bordeaux.fr),

<sup>2</sup> Department of Computer Science, Durham University, UK

[{konrad.dabrowski,matthew.johnson2,daniel.paulusma}@durham.ac.uk](mailto:{konrad.dabrowski,matthew.johnson2,daniel.paulusma}@durham.ac.uk),

<sup>3</sup> IRIF & Université Paris Diderot, Paris France [feghali@irif.fr](mailto:feghali@irif.fr)

**Abstract.** The NEAR-BIPARTITENESS problem is that of deciding whether or not the vertices of a graph can be partitioned into sets  $A$  and  $B$ , where  $A$  is an independent set and  $B$  induces a forest. The set  $A$  in such a partition is said to be an independent feedback vertex set. Yang and Yuan proved that NEAR-BIPARTITENESS is polynomial-time solvable for graphs of diameter 2 and NP-complete for graphs of diameter 4. We show that NEAR-BIPARTITENESS is NP-complete for graphs of diameter 3, resolving their open problem. We also generalise their result for diameter 2 by proving that even the problem of computing a minimum independent feedback vertex is polynomial-time solvable for graphs of diameter 2.

**Keywords:** near-bipartite graphs, independent feedback vertex set, diameter, computational complexity

## 1 Introduction

A graph is *near-bipartite* if its vertex set can be partitioned into sets  $A$  and  $B$ , where  $A$  is an independent set and  $B$  induces a forest. The set  $A$  is said to be an *independent feedback vertex set* and the pair  $(A, B)$  is said to be a *near-bipartite decomposition*. This leads to the following two related decision problems.

NEAR-BIPARTITENESS

*Instance:* a graph  $G$ .

*Question:* is  $G$  near-bipartite (that is, does  $G$  have an independent feedback vertex set)?

INDEPENDENT FEEDBACK VERTEX SET

*Instance:* a graph  $G$  and an integer  $k \geq 0$ .

*Question:* does  $G$  have an independent feedback vertex set of size at most  $k$ ?

Setting  $k = n$  shows that the latter problem is more general than the first problem. Thus, if NEAR-BIPARTITENESS is NP-complete for some graph class, then so is INDEPENDENT FEEDBACK VERTEX SET, and if INDEPENDENT FEEDBACK VERTEX SET is polynomial-time solvable for some graph class, then so is NEAR-BIPARTITENESS.

Note that every near-bipartite graph is 3-colourable, that is, its vertices can be coloured with at most three colours such that no two adjacent vertices are coloured alike. The problems 3-COLOURING [11] and NEAR-BIPARTITENESS [6] (and thus INDEPENDENT FEEDBACK

---

\* This paper received support from EPSRC (EP/K025090/1), London Mathematical Society (41536), the Leverhulme Trust (RPG-2016-258) and Fondation Sciences Mathématiques de Paris. The hardness result (Theorem 4) of this paper has been announced in an extended abstract of the Proceedings of MFCS 2017 [4].

VERTEX SET) are NP-complete. However, their complexities do not necessarily coincide on special graph classes. Grötschel, Lovász and Schrijver [10] proved that COLOURING is polynomial-time solvable for perfect graphs even if the permitted number of colours  $k$  is part of the input. However, Brandstädt et al. [5] proved that NEAR-BIPARTITENESS remains NP-complete for perfect graphs. The same authors also showed that NEAR-BIPARTITENESS is polynomial-time solvable for  $P_4$ -free graphs.

Yang and Yuan [16] proved that NEAR-BIPARTITENESS also remains NP-complete for graphs of maximum degree 4. To complement their hardness result, Yang and Yuan [16] showed that every connected graph of maximum degree at most 3 is near-bipartite except the complete graph  $K_4$  on four vertices. This also follows from a more general result of Catlin and Lai [8]. Recently we gave a linear-time algorithm for finding an independent feedback vertex set in a graph of maximum degree at most 3 [4], and also proved that NEAR-BIPARTITENESS is NP-complete even for line graphs of maximum degree 4 [3]. It is also known that NEAR-BIPARTITENESS is NP-complete for planar graphs; this follows from a result of Dross, Montassier and Pinlou [9]; see the arXiv version of [4] for details.

Tamura, Ito and Zhou [15] proved that INDEPENDENT FEEDBACK VERTEX SET is NP-complete for planar bipartite graphs of maximum degree 4 (note that NEAR-BIPARTITENESS is trivial for bipartite graphs). They also proved that INDEPENDENT FEEDBACK VERTEX SET is linear-time solvable for graphs of bounded treewidth, chordal graphs and  $P_4$ -free graphs (the latter result generalising the result of [5] for NEAR-BIPARTITENESS on  $P_4$ -free graphs). In [3] we proved that finding a minimum independent feedback vertex set is polynomial-time solvable even for  $P_5$ -free graphs. We refer to [1,13] for FPT algorithms with parameter  $k$  for finding an independent feedback vertex set of size at most  $k$ .

The *distance* between two vertices  $u$  and  $v$  in a graph  $G$  is the length (number of edges) of a shortest path between  $u$  and  $v$ . The *diameter* of a graph  $G$  is the maximum distance between any two vertices in  $G$ . In addition to their results for graphs of bounded maximum degree, Yang and Yuan [16] proved that NEAR-BIPARTITENESS is polynomial-time solvable for graphs of diameter at most 2 and NP-complete for graphs of diameter at most 4. They asked the following question, which was also posed by Brandstädt et al. [5]:

*What is the complexity of NEAR-BIPARTITENESS for graphs of diameter 3?*

**Our Results.** We complete the complexity classifications of NEAR-BIPARTITENESS and INDEPENDENT FEEDBACK VERTEX SET for graphs of bounded diameter. In particular, we prove that NEAR-BIPARTITENESS is NP-complete for graphs of diameter 3, which answers the above question. We also prove that INDEPENDENT FEEDBACK VERTEX SET is polynomial-time solvable for graphs of diameter 2. This generalises the result of Yang and Yuan [16] for NEAR-BIPARTITENESS restricted to graphs of diameter 2.

**Theorem 1.** *Let  $k \geq 0$  be an integer.*

- (i) *If  $k \leq 2$ , then INDEPENDENT FEEDBACK VERTEX SET (and thus NEAR-BIPARTITENESS) is polynomial-time solvable for graphs of diameter  $k$ .*
- (ii) *If  $k \geq 3$ , then NEAR-BIPARTITENESS (and thus INDEPENDENT FEEDBACK VERTEX SET) is NP-complete for graphs of diameter  $k$ .*

We prove Theorem 1 (i) in Section 2. Yang and Yuan [16] proved their result for NEAR-BIPARTITENESS by giving a polynomial-time verifiable characterisation of the class of near-bipartite graphs of diameter 2. We use their characterisation as the starting point for our algorithm for INDEPENDENT FEEDBACK VERTEX SET. In fact our algorithm not only solves the decision problem but even finds a minimum independent feedback vertex set in a graph of diameter 2.

We prove Theorem 1 (ii) in Section 3 by using a construction of Mertzios and Spirakis [12], which they used to prove that 3-COLOURING is NP-complete for graphs of diameter 3.

The outline of their proof is straightforward: a reduction from 3-SATISFIABILITY that constructs, for any instance  $\phi$ , a graph  $H_\phi$  that is 3-colourable if and only if  $\phi$  is satisfiable. We reduce 3-SATISFIABILITY to NEAR-BIPARTITENESS for graphs of diameter 3 using the same construction, that is, we show that  $H_\phi$  is near-bipartite if and only if  $\phi$  is satisfiable. As such, our result is an observation about the proof of Mertzios and Spirakis, but, owing to the intricacy of  $H_\phi$ , this observation is non-trivial to verify. In Section 3 we therefore repeat the construction and describe our reduction in detail, though we rely on [12] where possible in the proof.

## 2 Independent Feedback Vertex Set for Diameter 2

In this section we show how to compute a minimum independent feedback vertex set of a graph of diameter 2 in polynomial time. As mentioned, our proof relies on a known characterisation of near-bipartite graphs of diameter 2 [16]. In order to explain this characterisation, we first need to introduce some terminology.

Let  $G = (V, E)$  be a graph and let  $X \subseteq V$ . Then the *2-neighbour set* of  $X$ , denoted by  $A_X$ , is the set that consists of all vertices in  $V \setminus X$  that have at least two neighbours in  $X$ . A set  $I \subseteq V$  is *independent* if no two vertices of  $I$  are adjacent. For  $u \in V$ , we let  $G - u$  denote the graph obtained from  $G$  after deleting the vertex  $u$  (and its incident edges). A graph is *complete bipartite* if its vertex set can be partitioned into two independent sets  $S$  and  $T$  such that there is an edge between every vertex of  $S$  and every vertex of  $T$ . If  $S$  or  $T$  has size 1, the graph is also called a *star*.

**Theorem 2 ([16]).** *A graph  $G = (V, E)$  of diameter 2 is near-bipartite if and only if one of the following two conditions holds:*

- (i) *there exists a vertex  $u$  such that  $G - u$  is bipartite; or*
- (ii) *there exists a set  $X$ ,  $4 \leq |X| \leq 5$ , such that  $(A_X, V \setminus A_X)$  is a near-bipartite decomposition.*

As noted in [16], Theorem 2 can be used to solve NEAR-BIPARTITENESS in polynomial time for graphs of diameter 2, as conditions (i) and (ii) can be checked in polynomial time. However, Theorem 2 does not tell us how to determine the size of a minimum independent feedback vertex set.

In order to find a minimum independent feedback vertex set, we will distinguish between the two cases of Theorem 2. This leads to two corresponding lemmas.

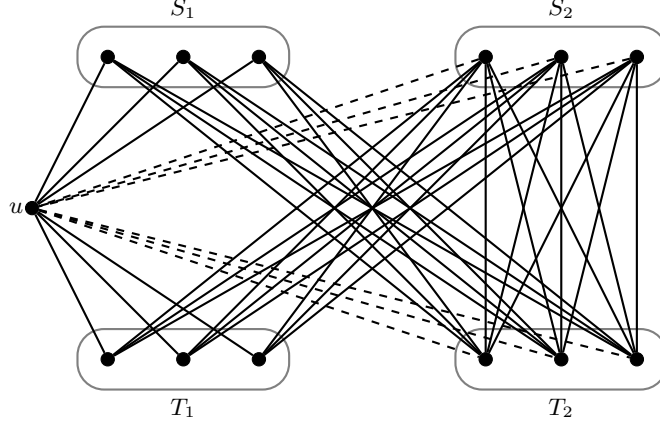
**Lemma 1.** *Let  $G = (V, E)$  be a near-bipartite graph of diameter 2 that contains a vertex  $u$  such that  $G - u$  is bipartite. Then it is possible to find a minimum independent feedback vertex set of  $G$  in polynomial time.*

*Proof.* Note that  $G$  is connected, as  $G$  has diameter 2. We can partition  $V \setminus \{u\}$  into four independent sets  $S_1, S_2, T_1, T_2$  (some of which might be empty) such that

- (i)  $S_1 \cup S_2$  and  $T_1 \cup T_2$  form bipartition classes of  $G - u$ ;
- (ii)  $u$  is adjacent to every vertex of  $S_1 \cup T_1$ ; and
- (iii)  $u$  is non-adjacent to every vertex of  $S_2 \cup T_2$ .

Moreover, as  $G$  has diameter 2, it follows that given a vertex of  $S_2$  (respectively,  $T_2$ ) and a vertex of  $T_1 \cup T_2$  (respectively,  $S_1 \cup S_2$ ), these two vertices must either be adjacent or have a common neighbour. As the latter is not possible, we deduce that

- (iv) every vertex of  $S_2$  is adjacent to every vertex of  $T_1 \cup T_2$ , and every vertex of  $T_2$  is adjacent to every vertex of  $S_1 \cup S_2$  (see also Fig. 1).



**Fig. 1.** The graph  $G$ , which consists of the vertex  $u$  and the independent sets  $S_1 \cup S_2$  and  $T_1 \cup T_2$ . Dashed lines indicate edges that are not present. Edges between vertices of  $S_1$  and vertices of  $T_1$  are not drawn, as such edges may or may not exist.

A (not necessarily proper) 2-colouring of the vertices of a graph is *good* if the vertices coloured 1 form an independent set and the vertices coloured 2 induce a forest. The set of vertices coloured 1 in a good 2-colouring is said to be a *1-set* and is, by definition, an independent feedback vertex of  $G$ . A good 2-colouring of  $G$  is *optimal* if its 1-set is of minimum possible size among all good 2-colourings. Our algorithm colours vertices one by one with colour 1 or 2 to obtain a number of good 2-colourings. We will establish that our approach ensures that at least one of our good 2-colourings is optimal. Therefore, as our algorithm finds different good 2-colourings, it only needs to remember the smallest 1-set seen so far. We note that  $G$  certainly has good 2-colourings as, for example, we can let either  $S_1 \cup S_2$  or  $T_1 \cup T_2$  be the set of vertices coloured 1.

We say that an edge is a *1-edge* if both its end-points have colour 1 and say that a cycle of  $G$  is a *2-cycle* if all its vertices have colour 2. Our algorithm will consist of a number of branches depending on the way we will colour the vertices of  $G$ . Whenever we detect a 1-edge or a 2-cycle in a branch, we can discard the branch as we know that we are not going to generate a good 2-colouring. Before we describe our algorithm, we first prove the following claim. Here, we say that an independent set  $I$  is a *twin-set* if every vertex of  $I$  has the same neighbourhood.

*Claim 1. Let  $I$  be a twin-set. In every optimal 2-colouring, at least  $|I| - 1$  vertices of  $I$  obtain the same colour.*

We prove Claim 1 as follows. If  $|I| = 1$ , the claim is trivial. Suppose  $|I| \geq 2$  and let  $J$  be the neighbourhood of the vertices of  $I$ . Note that  $J$  is non-empty since  $|I| \geq 2$  and  $G$  is connected. Let  $c$  be an optimal 2-colouring of  $G$ . If  $c$  gives colour 1 to a vertex of  $J$ , then every vertex of  $I$  must receive colour 2. Now suppose that  $c$  gives colour 2 to every vertex of  $J$ . If  $|J| = 1$ , then  $c$  colours every vertex of  $I$  with colour 2, as doing this will not create a 2-cycle. If  $|J| \geq 2$  then, in order to avoid a 2-cycle, at least  $|I| - 1$  vertices of  $I$  must be coloured 1. This proves Claim 1.

By (i), (iii), (iv), we find that  $S_2$  and  $T_2$  are twin-sets. Let  $Z$  be the set of isolated vertices in the subgraph of  $G$  induced by  $S_1 \cup T_1$ . Then by (i), (ii), (iv), the neighbourhood of every vertex in  $Z \cap S_1$  (respectively,  $Z \cap T_1$ ) is  $T_2 \cup \{u\}$  (respectively,  $S_2 \cup \{u\}$ ). So  $Z \cap S_1$  and  $Z \cap T_1$  are twin-sets.

We choose one vertex from each non-empty set in  $\{S_2, T_2, Z \cap S_1, Z \cap T_1\}$  and let  $W$  be the set of chosen vertices. Note that the choice of the vertices in  $W$  can be done arbitrarily,

since all four of these sets are twin-sets. We now branch by giving all vertices in  $S_2 \setminus W$  the same colour, all vertices in  $T_2 \setminus W$  the same colour, all vertices in  $(Z \cap S_1) \setminus W$  the same colour and all vertices in  $(Z \cap T_1) \setminus W$  the same colour. We then branch by colouring the at most four vertices of  $W$  with every possible combination of colours. Hence the total number of branches is at most  $2^8$ . We discard any branch that yields a 1-edge or 2-cycle. Let  $S'_1 = S_1 \setminus Z$  and  $T'_1 = T_1 \setminus Z$ . For each remaining branch we try to colour the remaining vertices of  $G$ , which are all in  $S'_1 \cup T'_1 \cup \{u\}$ , and keep track of any minimum 1-set found. In the end we return a 1-set of minimum size (recall that  $G$  has at least two 1-sets).

For any remaining branch we do as follows. We first give colour 1 to  $u$ . Then every vertex of  $S'_1 \cup T'_1$  must get colour 2. If this does not yield a 1-edge or 2-cycle, we obtain a 1-set, which we remember if it is the smallest one found so far.

We now give colour 2 to  $u$ . If  $u$  was the only remaining vertex, we check for the presence of a 1-edge or a 2-cycle, and if none is present, we remember the 1-set found if it is the smallest one found so far. Otherwise, we let  $D_1, \dots, D_r$  for some integer  $r \geq 1$  be the connected components of the (bipartite) graph induced by  $S'_1 \cup T'_1$ . As these vertices do not belong to  $Z$ , each  $D_i$  contains at least one edge. Moreover, each  $D_i$  is bipartite. For  $i \in \{1, \dots, r\}$ , we denote the two non-empty bipartition classes of  $D_i$  by  $D_i^1$  and  $D_i^2$  such that  $|V(D_i^1)| \leq |V(D_i^2)|$ . The following claim is crucial.

*Claim 2. For  $i \in \{1, \dots, r\}$ , we must either colour all vertices of  $D_i^1$  with colour 1 and all vertices of  $D_i^2$  with colour 2, or vice versa.*

We prove Claim 2 as follows. Suppose that  $D_i^1$  contains a vertex with the same colour as a vertex of  $D_i^2$ . As  $D_i$  is connected and bipartite, this means that  $D_i$  contains an edge  $vw$  whose end-vertices are either both coloured 1 or coloured 2. In the first case, we obtain a 1-edge. In the second case the vertices  $u, v$  and  $w$  form a 2-cycle in  $G$ . Hence we must use colours 1 and 2 for different partition classes of  $D_i$ . This proves Claim 2.

We now proceed as follows. First suppose that  $S_2 \cup T_2$  is non-empty. If we coloured a vertex in  $S_2$  (respectively  $T_2$ ) with colour 1, then every vertex in  $T'_1$  (respectively  $S'_1$ ) must be coloured 2 and therefore every vertex in  $S'_1$  (respectively  $T'_1$ ) must be coloured 1 by Claim 2. Again, in this case we discard the branch if a 1-edge or 2-cycle is found; otherwise we remember the corresponding 1-set if it is the best set found so far. In every other case, we must have coloured every vertex of non-empty set  $S_2 \cup T_2$  with colour 2. Without loss of generality, assume that there is a vertex  $s \in S_2$  that is coloured 2. Then at most one vertex of  $T'_1$  may have colour 2, as otherwise we obtain a 2-cycle by involving the vertices  $s$  and  $u$ . We branch by guessing this vertex and then colouring it either 1 or 2, while assigning colour 1 to all other vertices of  $T'_1$ . Then the only vertices with no colour yet are in  $S'_1$ , but their colour is determined by the colours of the vertices in  $T'_1$  due to Claim 2.

We are left to deal with the case where  $S_2 \cup T_2 = \emptyset$ . Claim 2 tells us that we must either give every vertex of  $D_i^1$  colour 1 and every vertex of  $D_i^2$  colour 2, or vice versa. For  $i \in \{1, \dots, r\}$  we give colour 1 to every vertex of every  $D_i^1$ ; as  $|V(D_i^1)| \leq |V(D_i^2)|$ , this is the best possible good 2-colouring for this branch.

The correctness of our algorithm follows from the fact that we distinguish all possible cases and find a best possible good 2-colouring (if one exists) in each case. Note that it takes polynomial time to find the sets  $S_1, S_2, T_1$  and  $T_2$ . Moreover, the number of branches is  $O(n)$  and each branch can be processed in polynomial time, as we only need to search for a 1-edge or 2-cycle. Hence our algorithm runs in polynomial time.  $\square$

**Lemma 2.** *Let  $G = (V, E)$  be a near-bipartite graph of diameter 2 that contains no vertex  $u$  such that  $G - u$  is bipartite. Then it is possible to find a minimum independent feedback vertex set of  $G$  in polynomial time.*

*Proof.* As  $G$  is near-bipartite, it has an independent feedback vertex set. Let  $A$  be a minimum independent feedback vertex set. We claim that  $G$  contains a set of vertices  $X$  of

size  $4 \leq |X| \leq 5$  such that  $A_X = A$ . This would immediately give us a polynomial-time algorithm. Indeed, it would suffice to check, for every set  $X$  of size  $4 \leq |X| \leq 5$ , whether  $(A_X, V \setminus A_X)$  is a near-bipartite decomposition and to return a set  $A_X$  of minimum size that satisfies this condition. This takes polynomial time.

To prove the above claim we will follow the same line of reasoning as in the proof of Theorem 2. However, our arguments are slightly different, as we need to prove a stronger statement.

Let  $B = V \setminus A$  and let  $F$  be the subgraph of  $G$  induced by  $B$ . By definition,  $F$  is a forest, so all of its connected components are trees.

We will first consider the case where  $F$  has a connected component  $T$  of diameter at least 3. Let  $P$  be a longest path in the tree  $T$  on vertices  $v_1, \dots, v_p$  in that order. As  $T$  has diameter 3, we find that  $p \geq 4$ . If  $p \leq 5$ , then we let  $X = \{v_1, \dots, v_p\}$ . If  $p \geq 6$ , then we let  $X = \{v_1, v_2, v_{p-1}, v_p\}$ . We will show that  $A = A_X$ . Let  $u \in A$ . As  $G$  has diameter 2 and  $A$  is an independent set,  $u$  is adjacent to  $v_1$  or to a neighbour  $v^*$  of  $v_1$  in  $B$ . In the latter case, if  $v^* \neq v_2$  then  $v^*$  must have a neighbour in  $\{v_2, \dots, v_p\}$ , otherwise we have found a path that is longer than  $P$ , but in this case  $B$  contains a cycle, a contradiction. Hence,  $u$  has at least one neighbour in  $\{v_1, v_2\}$ , and similarly,  $u$  has at least one neighbour in  $\{v_{p-1}, v_p\}$ . So  $A \subseteq A_X$ . Now suppose  $u \in A_X$ . Note that  $u \neq v_3$  due to our choice of  $X$ . Then the subgraph of  $G$  induced by  $V(P) \cup \{u\}$  contains a cycle. Hence  $u$  must belong to  $A$ . So  $A_X \subseteq A$ . We conclude that  $A = A_X$ .

We now consider the case where every connected component of  $F$  has diameter at most 2. Such components are either isolated vertices or stars (we say that the latter components are *star-components* and that their non-leaf vertex is the *star-centre*; if such a component consists of a single edge, we arbitrarily choose one of them to be the star-centre). If  $F$  contains no star-components, then  $G$  is bipartite and therefore  $G - u$  is bipartite for every vertex  $u$ , a contradiction. If  $F$  contains exactly one star-component, then by choosing  $u$  to be the star-centre we again find that  $G - u$  is bipartite. Hence  $F$  contains at least two star-components  $D_1$  and  $D_2$ . For  $i = 1, 2$ , let  $v_i$  be the star-centre and let  $w_i$  be a leaf in  $D_i$ .

We choose  $X = \{v_1, v_2, w_1, w_2\}$  and show that  $A = A_X$ . Let  $u \in A$ . As  $G$  has diameter 2 and  $A$  is an independent set,  $u$  is either adjacent to  $w_1$  or to a neighbour of  $w_1$  in  $B$ . If this neighbour is not  $v_1$ , then  $D_1$  is not a star-component, a contradiction. Hence,  $u$  has at least one neighbour in  $\{v_1, w_1\}$ , and similarly,  $u$  has at least one neighbour in  $\{v_2, w_2\}$ . So  $A \subseteq A_X$ . Now suppose  $u \in A_X$ . Then  $X \cup \{u\}$  induces either a connected subgraph of  $G$  that contains both  $D_1$  and  $D_2$  (and is therefore not a star-component) or a subgraph with a cycle. Hence  $u$  must belong to  $A$ . So  $A_X \subseteq A$ . We conclude that  $A = A_X$ . This completes the proof of our claim and thus the proof of the lemma.  $\square$

We are now ready to prove the main result of this section.

**Theorem 3.** *The problem of finding a minimum independent feedback vertex set of a graph of diameter 2 can be solved in polynomial time.*

*Proof.* Let  $G$  be an  $n$ -vertex graph of diameter 2. We first check in polynomial time whether  $G$  contains a vertex  $u$  such that  $G - u$  is bipartite. If so, then we apply Lemma 1. If not, then we check in polynomial time whether  $G$  contains a set  $X$  of size  $4 \leq |X| \leq 5$  such that  $(A_X, V \setminus A_X)$  is a near-bipartite decomposition. If so, then  $G$  is near-bipartite and we apply Lemma 2. If not, then  $G$  is not near-bipartite due to Theorem 2.  $\square$

We note that the running time of the algorithm in Theorem 3 is determined by the time it takes to find and process each set  $X$  of size  $4 \leq |X| \leq 5$ . This takes  $O(n^7)$  time, as checking the existence of a set  $X$  takes  $O(n^5)$  time using brute force, determining the 2-neighbour set  $A_X$  takes  $O(n)$  time and checking if  $(A_X, V \setminus A_X)$  is a near-bipartite decomposition takes  $O(n^2)$  time.

### 3 Near-Bipartiteness for Diameter 3

In this section we prove that NEAR-BIPARTITENESS is NP-complete for graphs of diameter 3. In order to prove this, we use a construction of Mertzios and Spirakis [12]. To introduce this construction, we first consider the constraint graph  $J$  defined in Figure 2.

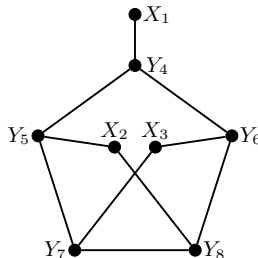


Fig. 2. The constraint graph  $J$ .

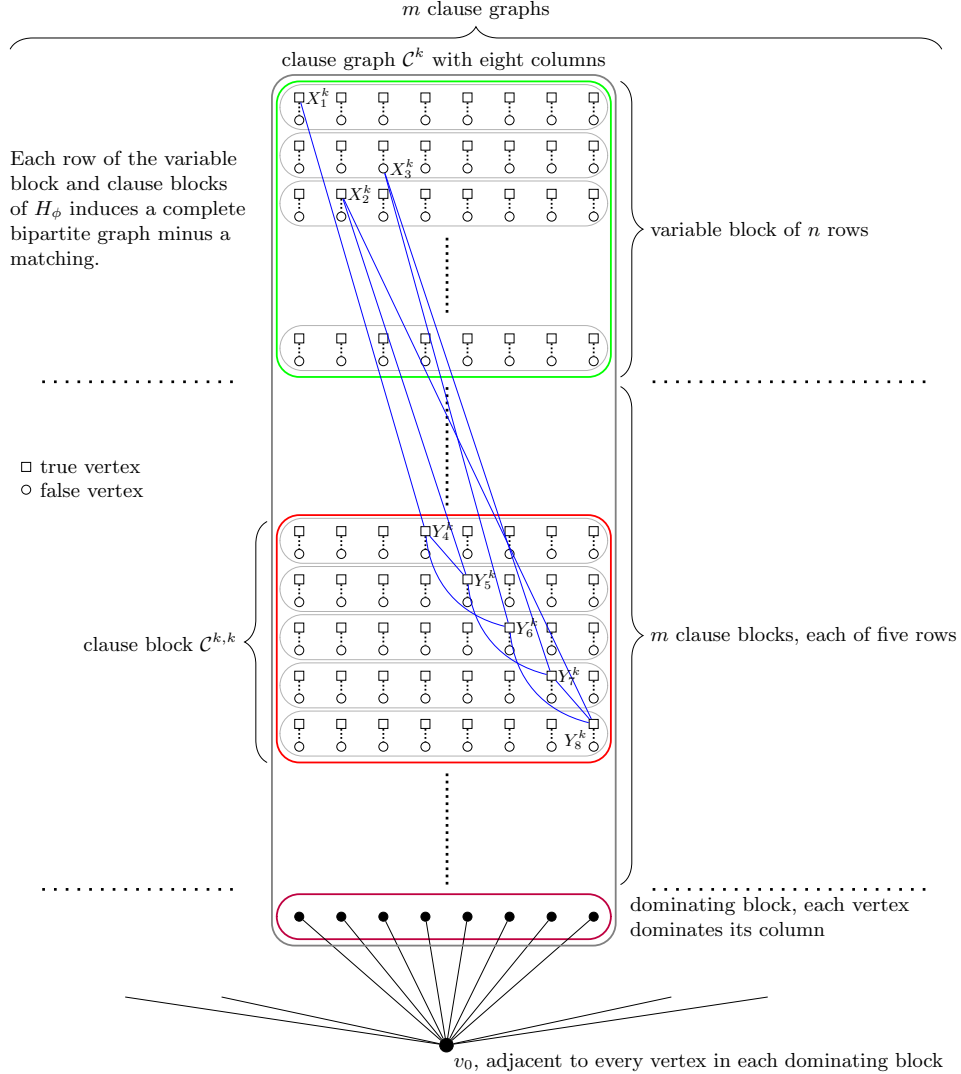
**Lemma 3.** *Let  $X$  be a subset of  $\{X_1, X_2, X_3\} \subset V(J)$  containing at most two vertices. Then there exists a near-bipartite decomposition  $(A, B)$  of  $J$  such that, for  $1 \leq p \leq 3$ ,  $X_p \in A$  if and only if  $X_p \in X$ .*

*Proof.* Noting the automorphic equivalence of  $X_2$  and  $X_3$ , it is sufficient to consider the following two cases. If  $X$  is a subset of  $\{X_1, X_2\}$ , let  $A = X \cup \{Y_6, Y_7\}$ . If  $X = \{X_2, X_3\}$ , let  $A = \{X_2, X_3, Y_4\}$ .  $\square$

Notice that there is no near-bipartite decomposition  $(A, B)$  of  $J$  with  $\{X_1, X_2, X_3\} \subseteq A$ . Combined with the above lemma, this gives an idea of how this will be used later. The vertices  $X_1$ ,  $X_2$  and  $X_3$  will represent literals in a clause of an instance of 3-SAT and membership of  $A$  will indicate that a literal is false: thus  $A$  can be extended to a near-bipartite decomposition except when every literal is false. (In [12], a weaker result was shown: one can always find a 3-colouring of  $J$  such that members of a chosen *proper* subset of  $\{X_1, X_2, X_3\}$  belong to the same class and excluded members do not belong to that class.)

Let  $\phi$  be an instance of 3-SAT with  $m$  clauses  $C_1, \dots, C_m$  and  $n$  variables  $v_1, \dots, v_n$ . We may assume that each clause has three distinct literals. For a clause  $C_k$  in  $\phi$ , we describe a *clause graph*  $\mathcal{C}^k$ , illustrated within Figure 3. We think of  $\mathcal{C}^k$  as an array of  $n + 5m + 1$  rows and eight columns. In each row except the last, every (row, column) position contains exactly two vertices, which we refer to as the *true vertex* and the *false vertex*, and we say that these two vertices are *mates*. The first  $n$  rows form the *variable block* of the graph and we think of row  $i$  as representing the variable  $v_i$ . The next  $5m$  rows are made up of  $m$  *clause blocks*  $\mathcal{C}^{k,1}, \mathcal{C}^{k,2}, \dots, \mathcal{C}^{k,m}$ , each of five rows. Every true vertex of the variable and clause blocks is joined by an edge to every false vertex in the same row except its mate. Hence the vertices of each row induce a complete bipartite graph minus a matching. In the final row, each column contains a single vertex, and each of these vertices is joined by an edge to every other vertex in the same column. We call this row the *dominating block*. We complete the definition of the clause graph by describing how we add further edges so that it contains the constraint graph  $J$  as an induced subgraph. Let the literals of  $C_k$  be  $x_{\ell_1}, x_{\ell_2}, x_{\ell_3}$ . We choose vertices from the first three columns of the variable block of  $\mathcal{C}^k$  that we will denote  $X_1^k, X_2^k, X_3^k$  to represent the literals. If  $x_{\ell_p}$  is the variable  $v_i$ , then we choose as  $X_p^k$  a vertex from row  $i$  and column  $p$ , and choose the true vertex if the literal is positive

and the false vertex if the literal is a negated variable. For  $p \in \{4, \dots, 8\}$ , let  $Y_p^k$  be the true vertex from the  $(p-3)$ th row and  $p$ th column of the clause block  $\mathcal{C}^{k,k}$ . Finally add the ten edges  $\{X_1^k Y_4^k, X_2^k Y_5^k, X_2^k Y_8^k, X_3^k Y_6^k, X_3^k Y_7^k, Y_4^k Y_5^k, Y_4^k Y_6^k, Y_5^k Y_7^k, Y_6^k Y_8^k, Y_7^k Y_8^k\}$  so that  $\{X_1^k, X_2^k, X_3^k, Y_4^k, Y_5^k, Y_6^k, Y_7^k, Y_8^k\}$  induces the constraint graph  $J$ . Note that each vertex in the dominating block has only one neighbour in the constraint graph.



**Fig. 3.** The graph  $H_\phi$  with the focus on one of the constituent clause graphs  $\mathcal{C}^k$ , where  $\mathcal{C}^k = \{v_1 \vee v_3 \vee \bar{v}_2\}$ . The blue edges illustrate the induced constraint graph.

We now define the graph  $H_\phi$ . It contains:

- the disjoint union of clause graphs  $\mathcal{C}^k$ ,  $1 \leq k \leq m$  (we think of the clause graphs as being arranged side-by-side, so that they form an array of  $n + 5m + 1$  rows and  $8m$  columns),
- edges from each true vertex of each clause graph to each false vertex in the same row of other clause graphs, and



- an additional vertex  $v_0$  joined to each vertex in the dominating block of each clause graph.

Note that each column of  $H_\phi$  contains exactly one vertex that is in a constraint graph  $J$  and the only rows that contain more than one such vertex are those in the variable block.

**Lemma 4** ([12, Lemma 2]). *For an instance  $\phi$  of 3-SAT,  $H_\phi$  has diameter 3.*

Note that in [12], Lemma 2 proves the bound on the diameter for a graph that is a spanning subgraph of  $H_\phi$  which is, of course, sufficient for an upper bound for the diameter of  $H_\phi$  and it is easy to see that the diameter is not less than 3. We note also that  $H_\phi$  does not contain any triangles or any vertices that are siblings (two vertices are siblings if the neighbourhood of one is a subset of the neighbourhood of the other) so NEAR-BIPARTITENESS is also NP-complete for such instances.

**Theorem 4.** NEAR-BIPARTITENESS is NP-complete for graphs of diameter at most 3.

*Proof.* We prove that 3-SAT can be polynomially-reduced to NEAR-BIPARTITENESS by showing that  $\phi$  is satisfiable if and only if  $H_\phi$  has a near-bipartite decomposition  $(A, B)$ .

( $\Rightarrow$ ) Suppose that  $\phi$  has a satisfying assignment. Let  $v_0$  be in  $A$ , and let the vertices of all the dominating blocks be in  $B$ . If the variable  $v_i$  is true, then let  $B$  contain all the true vertices of row  $i$  of the variable blocks of each clause graph. Otherwise let  $B$  contain the false vertices. In each case, let  $A$  contain the mates of these vertices. Consider the constraint graph that is an induced subgraph of each clause graph. The vertices  $X_1$ ,  $X_2$  and  $X_3$  have been assigned to either  $A$  or  $B$  with at most two, representing false literals, belonging to  $A$ . By Lemma 3, we can assign the remaining vertices of the subgraph (which are all true vertices of clause blocks) to  $A$  and  $B$  such that on the subgraph they form a near-bipartite decomposition. When we assign a true vertex of a clause block to  $A$  or  $B$ , we assign all other true vertices in the same row of  $H_\phi$  to the same set and assign their mates to the other set. As each row of the clause blocks contains only one vertex in a constraint graph, this process assigns every vertex in  $H_\phi$  to exactly one of  $A$  and  $B$ , and we have assigned every vertex of  $H_\phi$  to  $A$  or  $B$ .

It is immediately clear that  $A$  is an independent set. We must show that  $B$  contains no cycles. We know that  $B$  contains all the vertices of the dominating blocks and, in each row, either all the true vertices or all the false vertices. Thus if  $B$  contains a cycle then all the vertices of the cycle belong to the same clause graph (the only edges going between distinct clause graphs are those joining true vertices to false vertices in the same row). Let  $G_B$  be a subgraph of a clause graph induced by vertices of  $B$ . Then each true and false vertex not in the constraint graph has degree 1 (due to the edge joining it to the dominating block), and each vertex in the dominating block has at most one neighbour with degree more than 1 (since it only has one neighbour in the constraint graph). Thus if  $G_B$  contains a cycle then it belongs to the constraint graph, contradicting how  $A$  and  $B$  were chosen.

( $\Leftarrow$ ) Suppose  $A$  and  $B$  form a near-bipartite decomposition of  $H_\phi$ . Then  $B$  can be decomposed into two independent sets, and these, along with  $A$ , can be considered a 3-colouring. In [12, Theorem 5], it is shown that if  $H_\phi$  has a 3-colouring, then  $\phi$  is satisfiable.  $\square$

## 4 Conclusions

We completed the computational complexity classifications of NEAR-BIPARTITENESS and INDEPENDENT FEEDBACK VERTEX SET for graphs of diameter  $k$  for every integer  $k \geq 0$ . We showed that the complexity of both problems jumps from being polynomial-time solvable to NP-complete when  $k$  changes from 2 to 3.

We recall that near-bipartite graphs are 3-colourable. Interestingly, the complexity of 3-COLOURING for graphs of diameter  $k$  has not yet been settled, as there is one remaining case left, namely when  $k = 2$ . This is a notorious open problem, which has been frequently posed in the literature (see, for example, [2,7,12,14]). We note that the approach of solving NEAR-BIPARTITENESS and INDEPENDENT FEEDBACK VERTEX SET for graphs of diameter 2 does not work for 3-COLOURING. For instance, we cannot bound the size of the set  $X$  in Lemma 2 if we drop the condition that the union of two colour classes must induce a forest.

## References

1. A. Agrawal, S. Gupta, S. Saurabh, and R. Sharma. Improved algorithms and combinatorial bounds for independent feedback vertex set. *Proc. IPEC 2016, LIPIcs*, 63:2:1–2:14, 2017.
2. M. Bodirsky, J. Kára, and B. Martin. The complexity of surjective homomorphism problems – a survey. *Discrete Applied Mathematics*, 160(12):1680–1690, 2012.
3. M. Bonamy, K. K. Dabrowski, C. Feghali, M. Johnson, and D. Paulusma. Independent feedback vertex set for  $P_5$ -free graphs. *Proc. ISAAC 2017, LIPIcs*, 92:16:1–16:12, 2017.
4. M. Bonamy, K. K. Dabrowski, C. Feghali, M. Johnson, and D. Paulusma. Recognizing graphs close to bipartite graphs. *Proc. MFCS 2017, LIPIcs*, 83:70:1–70:14, 2017. Full version: arXiv:1707.09817.
5. A. Brandstädt, S. Brito, S. Klein, L. T. Nogueira, and F. Protti. Cycle transversals in perfect graphs and cographs. *Theoretical Computer Science*, 469:15–23, 2013.
6. A. Brandstädt, V. B. Le, and T. Szymczak. The complexity of some problems related to graph 3-colorability. *Discrete Applied Mathematics*, 89(1–3):59–73, 1998.
7. H. Broersma, F. V. Fomin, P. A. Golovach, and D. Paulusma. Three complexity results on coloring  $P_k$ -free graphs. *European Journal of Combinatorics*, 34(3):609–619, 2013.
8. P. A. Catlin and H.-J. Lai. Vertex arboricity and maximum degree. *Discrete Mathematics*, 141(1–3):37–46, 1995.
9. F. Dross, M. Montassier, and A. Pinlou. Partitioning a triangle-free planar graph into a forest and a forest of bounded degree. *European Journal of Combinatorics*, 66:81–94, 2017.
10. M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. *Annals of Discrete Mathematics*, 21:325–356, 1984.
11. L. Lovász. Coverings and coloring of hypergraphs. *Congressus Numerantium*, VIII:3–12, 1973.
12. G. B. Mertzios and P. G. Spirakis. Algorithms and almost tight results for 3-colorability of small diameter graphs. *Algorithmica*, 74(1):385–414, 2016.
13. N. Misra, G. Philip, V. Raman, and S. Saurabh. On parameterized independent feedback vertex set. *Theoretical Computer Science*, 461:65–75, 2012.
14. D. Paulusma. Open problems on graph coloring for special graph classes. *Proc. WG 2015, LNCS*, 9224:16–30, 2016.
15. Y. Tamura, T. Ito, and X. Zhou. Algorithms for the independent feedback vertex set problem. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E98-A(6):1179–1188, 2015.
16. A. Yang and J. Yuan. Partition the vertices of a graph into one independent set and one acyclic set. *Discrete Mathematics*, 306(12):1207–1216, 2006.