



ELSEVIER

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

More results on weighted independent domination [☆]Vadim Lozin ^{a,*}, Dmitriy Malyshev ^b, Raffaele Mosca ^c, Viktor Zamaraev ^a^a Mathematics Institute, University of Warwick, Coventry CV4 7AL, UK^b National Research University Higher School of Economics, 25/12 Bolshaya Pecherskaya Ulitsa, 603155, Nizhny Novgorod, Russia^c Dipartimento di Economia, Università degli Studi "G. D'Annunzio", Pescara 65121, Italy

ARTICLE INFO

Article history:

Received 17 May 2017

Accepted 7 August 2017

Available online 25 August 2017

Communicated by D.-Z. Du

Keywords:

Independent domination

Polynomial-time algorithm

ABSTRACT

Weighted independent domination is an NP-hard graph problem, which remains computationally intractable in many restricted graph classes. In particular, the problem is NP-hard in the classes of sat-graphs and chordal graphs. We strengthen these results by showing that the problem is NP-hard in a proper subclass of the intersection of sat-graphs and chordal graphs. On the other hand, we identify two new classes of graphs where the problem admits polynomial-time solutions.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

INDEPENDENT DOMINATION is the problem of finding in a graph an inclusionwise maximal independent set of minimum cardinality. This is one of the hardest problems of combinatorial optimisation and it remains difficult under substantial restrictions. In particular, it is NP-hard for so-called sat-graphs, where the problem is equivalent to SATISFIABILITY [19]. It is also NP-hard for planar graphs, triangle-free graphs, graphs of vertex degree at most 3 [3], line graphs [18], chordal bipartite graphs [7], etc.

The weighted version of the problem (abbreviated WID) deals with vertex-weighted graphs and asks to find an inclusionwise maximal independent set of minimum total weight. This version is provenly harder, as it remains NP-hard even for chordal graphs [5], where INDEPENDENT DOMINATION can be solved in polynomial time [8]. In the present paper, we strengthen two NP-hardness results by showing that WID is NP-hard in a proper subclass of the intersection of sat-graphs and chordal graphs.

On the positive side, it is known that the problem is polynomial-time solvable for interval graphs, permutation graphs [4], graphs of bounded clique-width [6], etc.

Let us observe that all classes mention above are hereditary, i.e. closed under taking induced subgraphs. It is well-known (and not difficult to see) that a class of graphs is hereditary if and only if it can be characterised in terms of minimal forbidden induced subgraphs. Unfortunately, not much is known about efficient solutions for the WID problem on graph classes defined by *finitely many* forbidden induced subgraphs. Among rare examples of this type, let us mention cographs and split graphs.

[☆] Extended abstract of this paper appeared in the proceedings of WG 2017 – the 43rd International Workshop on Graph-Theoretic Concepts in Computer Science [14].

* Corresponding author.

E-mail addresses: V.Lozin@warwick.ac.uk (V. Lozin), dmalishev@hse.ru (D. Malyshev), R.Mosca@unich.it (R. Mosca), V.Zamaraev@warwick.ac.uk (V. Zamaraev).

- A *cograph* is a graph in which every induced subgraph with at least two vertices is either disconnected or the complement of a disconnected graph. The cographs are precisely P_4 -free graphs, i.e. graphs containing no induced P_4 . In the case of cographs, the problem can be solved efficiently by means of modular decomposition.
- A *split graph* is a graph whose vertices can be partitioned into a clique and an independent set. In terms of forbidden induced subgraphs, the split graphs are the graphs which are free of $2K_2, C_4$ and C_5 . The only available way to solve WID efficiently for a split graph is to examine all its inclusionwise maximal independent sets, of which there are polynomially many.

The class of sat-graphs, mentioned earlier, consists of graphs whose vertices can be partitioned into a clique and a graph of vertex degree at most 1. Therefore, sat-graphs form an extension of split graphs. With this extension the complexity status of the problem jumps from polynomial-time solvability to NP-hardness. In the present paper, we study two other extensions of split graphs and show polynomial-time solvability in both of them.

The first of them deals with the class of $(P_5, \overline{P_5})$ -free graphs, which also extends the cographs. From an algorithmic point of view, this extension is resistant to any available technique. To crack the puzzle for $(P_5, \overline{P_5})$ -free graphs, we develop a new decomposition scheme combining several algorithmic tools. This enables us to show that the WID problem can be solved for $(P_5, \overline{P_5})$ -free graphs in polynomial time.

The second extension of split graphs studied in this paper deals with the class of $(P_5, \overline{P_3 + P_2})$ -free graphs. To solve the problem in this case, we develop a tricky reduction allowing us to reduce the problem to the first class.

Let us emphasise that in both cases the presence of P_5 among the forbidden graphs is necessary, because each of $\overline{P_5}$ and $\overline{P_3 + P_2}$ contains a C_4 and by forbidding C_4 alone we obtain a class where the problem is NP-hard [3]. Whether the presence of P_5 among the forbidden graphs is sufficient for polynomial-time solvability of WID is a big open question. For the related problem of finding a maximum weight independent set (WIS), this question was answered only recently [12] after several decades of attacking the problem on subclasses of P_5 -free graphs (see e.g. [2,9,11]). In particular, prior to solving the problem for P_5 -free graphs, it was solved for (P_5, H) -free graphs for all graphs H with at most 5 vertices, except for $H = C_5$.

WID is a more stubborn problem, as it remains NP-hard in many classes where WIS can be solved in polynomial time, such as line graphs, chordal graphs, bipartite graphs, etc. In [13], the problem was solved in polynomial time for many subclasses of P_5 -free graphs, including (P_5, H) -free graphs for all graphs H with at most 5 vertices, except for $H = \overline{P_5}$, $H = \overline{P_3 + P_2}$ and $H = C_5$. In the present paper, we solve the first two of them, leaving the case of (P_5, C_5) -free graphs open. We believe that WID in (P_5, C_5) -free graphs is polynomially equivalent to WID in P_5 -free graphs. Determining the complexity status of the problem in both classes is a challenging open question. We discuss this and related open questions in the concluding section of the paper.

The rest of the paper is organised as follows. In the remainder of the present section, we introduce basic terminology and notation. In Section 3 we solve the problem for $(P_5, \overline{P_5})$ -free graphs, and in Section 4 we solve it for $(P_5, \overline{P_3 + P_2})$ -free graphs.

All graphs in this paper are finite, undirected, without loops and multiple edges. The vertex set and the edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively. A subset $S \subseteq V(G)$ is

- *independent* if no two vertices of S are adjacent,
- a *clique* if every two vertices of S are adjacent,
- *dominating* if every vertex not in S is adjacent to a vertex in S .

For a vertex-weighted graph G with a weight function w , by $id_w(G)$ we denote the minimum weight of an independent dominating set in G .

If v is a vertex of G , then $N(v)$ is the *neighbourhood* of v (i.e. the set of vertices adjacent to v) and $V(G) \setminus N(v)$ is the *antineighbourhood* of v . We say that v is *simplicial* if its neighbourhood is a clique, and v is *antisimplicial* if its antineighbourhood is an independent set.

Let S be a subset of $V(G)$. We say that a vertex $v \in V(G) \setminus S$ *dominates* S if $S \subseteq N(v)$. Also, v *distinguishes* S if v has both a neighbour and a non-neighbour in S . By $G[S]$ we denote the subgraph of G induced by S and by $G - S$ the subgraph $G[V \setminus S]$. If S consists of a single element, say $S = \{v\}$, we write $G - v$, omitting the brackets.

If G is a connected graph but $G - S$ is not, then S is a *separator* (also known as a cut-set). A *clique separator* is a separator which is also a clique.

As usual, P_n, C_n and K_n denote a chordless path, a chordless cycle and a complete graph on n vertices, respectively. Given two graphs G and H , we denote by $G + H$ the disjoint union of G and H , and by mG the disjoint union of m copies of G .

We say that a graph G contains a graph H as an induced subgraph if H is isomorphic to an induced subgraph of G . Otherwise, G is H -free.

A class \mathcal{Z} of graphs is hereditary if it is closed under taking induced subgraphs, i.e. if $G \in \mathcal{Z}$ implies that every induced subgraph of G belongs to \mathcal{Z} . It is well-known that \mathcal{Z} is hereditary if and only if graphs in G do not contain induced subgraphs from a set M , in which case we say that M is the set of forbidden induced subgraphs for \mathcal{Z} .

For an initial segment of natural numbers $\{1, 2, \dots, n\}$ we will often use the notation $[n]$.

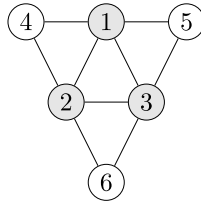


Fig. 1. Graph Sun_3 .

2. An NP-hardness result

As we mentioned in the introduction, the WID problem is NP-hard in the classes of sat-graphs and chordal graphs. A graph is *chordal* if it is (C_4, C_5, C_6, \dots) -free. A graph G is called a *sat-graph* if there exists a partition $A \cup B = V(G)$ such that

1. A is a clique (possibly, $A = \emptyset$);
2. $G[B]$ is an induced matching, i.e. an induced 1-regular graph (possibly, $B = \emptyset$);
3. there are no triangles (a, b, b') , where $a \in A$ and $b, b' \in B$.

We shall refer to the pair (A, B) as a *sat-partition* of G .

Below we show that WID is NP-hard in the class of (C_4, Sun_3) -free sat-graphs, where Sun_3 is the graph shown in Fig. 1. Since cycles C_k with $k \geq 5$ are not sat-graphs (which is easy to see), this class also is a subclass of chordal graphs. Moreover, Sun_3 is both a sat-graph and a chordal graph. Therefore, (C_4, Sun_3) -free sat-graphs form a proper subclass of the intersection of sat-graphs and chordal graphs.

Before we prove the main result of this section, let us make the following useful observation.

Observation 1. Let G be a sat-graph with a sat-partition (A, B) . If G contains Sun_3 as an induced subgraph, then $1, 2, 3 \in A$ and $4, 5, 6 \in B$.

Theorem 1. The WID problem is NP-hard in the class of (C_4, Sun_3) -free sat-graphs.

Proof. We prove the theorem by transforming the decision version of the MINIMUM DOMINATING SET problem in (C_3, C_4, C_5, C_6) -free graphs to the WID problem in (C_4, Sun_3) -free graphs. Since the former problem is NP-complete (see [10]), this will prove that the latter is NP-hard.

For an n -vertex graph $G = (V, E)$ let us define the graph $G' = (V', E')$ with vertex set $V' = \{v_1, v_2, v_3 : v \in V\}$ and edge set $E' = \{(v_1, v_2), (v_2, v_3) : v \in V\} \cup \{(w_2, v_3), (w_3, v_2) : (w, v) \in E\} \cup \{(w_3, v_3) : w, v \in V, u \neq v\}$.

Fig. 2 illustrates the transformation of P_4 into P'_4 . It is easy to see that for every graph G , the graph G' is a sat-graph. Moreover, it is C_4 -free, i.e. G' is a chordal graph. Also using the fact that Sun_3 has the unique sat-partition (see Observation 1) it is not hard to check that if G' contains Sun_3 as an induced subgraph, then G has a cycle of length at most 6. Therefore, for any (C_3, C_4, C_5, C_6) -free graph G , the graph G' is a (C_4, Sun_3) -free sat-graph.

Further, for every $v \in V$ we assign weight 1 to vertex v_1 , weight 2 to vertex v_2 , and weight $2n$ to vertex v_3 .

Now, we claim that G has a dominating set of size at most k if and only if G' has an independent dominating set of total weight at most $n + k$. First, suppose G has a dominating set D of size at most k . Then $D' = \{v_2 : v \in D\} \cup \{v_1 : v \in V \setminus D\}$ is clearly an independent dominating set of G' with total weight at most $n + k$. On the other hand, suppose G' has an independent dominating set D' of total weight at most $n + k$. If $k \geq n$, then V is a dominating set of G of size at most k . If $k < n$, then D' cannot contain any of the vertices of weight $2n$ and hence D' is of the form $\{v_2 : v \in D\} \cup \{v_1 : v \in V \setminus D\}$ for some subset D of V . For any vertex $u \in V$, since u_3 is dominated in G' by some $v_2 \in D'$, we have that in G vertex u is dominated by $v \in D$. Hence, D is a dominating set of G . Moreover, the total weight of D' is $n + |D|$ implying that D is of size at most k . \square

3. WID in $(P_5, \overline{P_5})$ -free graphs

To solve the problem for $(P_5, \overline{P_5})$ -free graphs, we first develop a new decomposition scheme in Section 3.1.3 that combines modular decomposition (Section 3.1.1) and antineighborhood decomposition (Section 3.1.2). Then in Section 3.2 we apply it to $(P_5, \overline{P_5})$ -free graphs.

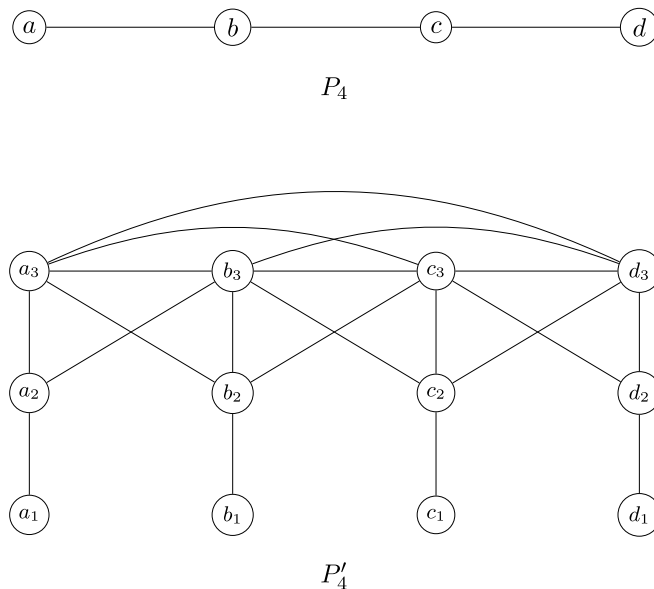


Fig. 2. Graphs P_4 (top) and P'_4 (bottom).

3.1. Graph decompositions

3.1.1. Modular decomposition

Let $G = (V, E)$ be a graph. A set $M \subseteq V$ is a *module* in G if no vertex outside of M distinguishes M . Obviously, $V(G)$, \emptyset and any vertex of G are modules and we call them *trivial*. A non-trivial module is also known as a *homogeneous set*. A graph without homogeneous sets is called *prime*. The notion of a prime graph plays a crucial role in *modular decomposition*, which allows to reduce various algorithmic and combinatorial problems in a hereditary class \mathcal{Z} to prime graphs in \mathcal{Z} (see e.g. [16] for more details on modular decomposition and its applications). In particular, it was shown in [3] that the WID problem can be solved in polynomial time in \mathcal{Z} whenever it is polynomially solvable for prime graphs in \mathcal{Z} .

In our solution, we will use homogeneous sets in order to reduce the problem from a graph G to two proper induced subgraphs of G as follows. Let $M \subset V$ be a homogeneous set in G . Denote by H the graph obtained from G by contracting M into a single vertex m (or equivalently, by removing all but one vertex m from M). We define the weight function w' on the vertices of H as follows: $w'(v) = w(v)$ for every $v \neq m$, and $w'(m) = id_w(G[M])$. Then it is not difficult to see that

$$id_w(G) = id_{w'}(H). \quad (1)$$

In other words, to solve the problem for G we first solve the problem for the subgraph $G[M]$, construct a new weighted graph H , and solve the problem for the graph H .

3.1.2. Antineighborhood decomposition

One of the simplest branching algorithms for the maximum weight independent set problem is based on the following obvious fact. For any graph $G = (V, E)$ and any vertex $v \in V$,

$$is_w(G) = \max\{is_w(G - N(v)), is_w(G - v)\},$$

where w is a weight function on the vertices of G , and $is_w(G)$ stands for the maximum weight of an independent set in G . We want to use a similar branching rule for the WID problem, i.e.

$$id_w(G) = \min\{id_w(G - N(v)), id_w(G - v)\}. \quad (2)$$

However, formula (2) is not necessarily true, because an independent dominating set in the graph $G - v$ is not necessarily dominating in the whole graph G . To overcome this difficulty, we introduce the following notion.

Definition 1. A vertex v is *permissible* if formula (2) is valid for v .

An obvious sufficient condition for a vertex to be permissible can be stated as follows: if every independent dominating set in $G - v$ contains at least one neighbour of v , then v is permissible.

Applying (2) to a permissible vertex v of G , we reduce the problem from G to two subgraphs $G - v$ and $G - N(v)$. Such a branching procedure results in a decision tree. In general, this approach does not provide a polynomial-time solution,

since the decision tree may have exponentially many nodes (subproblems). However, under some conditions this procedure may lead to a polynomial-time algorithm. In particular, this is true for graphs in hereditary classes possessing the following property.

Definition 2. A graph class \mathcal{G} has the *antineighborhood property* if there is a subclass $\mathcal{F} \subseteq \mathcal{G}$, and polynomial algorithms P , Q and R , such that

- (i) Given a graph G the algorithm P decides whether G belongs to \mathcal{F} or not;
- (ii) Q finds a permissible vertex v in any input graph $G \in \mathcal{G} \setminus \mathcal{F}$ such that the graph $G - N(v)$ induced by the antineighborhood of v belongs to \mathcal{F} ; we call v a *good vertex*;
- (iii) R solves the WID problem for (every induced subgraph of) any input graph from \mathcal{F} .

Directly from the definition we derive the following conclusion.

Theorem 2. Let \mathcal{G} be a hereditary class possessing the antineighborhood property. Then WID can be solved in polynomial time for graphs in \mathcal{G} .

3.1.3. Decomposition scheme

Let \mathcal{G} be a hereditary class such that the class \mathcal{G}_p of prime graphs in \mathcal{G} has the antineighborhood property. We define the decomposition procedure by describing the corresponding decomposition tree $T(G)$ for a graph $G = (V, E) \in \mathcal{G}$. In the description, we use notions and notations introduced in [Definition 2](#).

1. If G belongs to \mathcal{F} , then the node of $T(G)$ corresponding to G is a leaf.
2. If $G \notin \mathcal{F}$ and G has a homogeneous set M , then G is decomposed into subgraphs $G_1 = G[M]$ and $G_2 = G[(V \setminus M) \cup \{m\}]$ for some vertex m in M . The node of $T(G)$ corresponding to G is called a *homogeneous node*, and it has two children corresponding to G_1 and G_2 . These children are in turn the roots of subtrees representing possible decompositions of G_1 and G_2 .
3. If $G \notin \mathcal{F}$ and G has no homogeneous set, then G is prime and by the antineighborhood property of \mathcal{G}_p there exists a good vertex $v \in V$. Then G is decomposed into subgraphs $G_1 = G - N(v)$ and $G_2 = G - v$. The node of $T(G)$ corresponding to G is called an *antineighborhood node*, and it has two children corresponding to G_1 and G_2 . The graph G_1 belongs to \mathcal{F} and the node corresponding to G_1 is a leaf. The node corresponding to G_2 is the root of a subtree representing a possible decomposition of G_2 .

Lemma 3. Let G be an n -vertex graph in \mathcal{G} . Then the tree $T(G)$ contains $O(n^2)$ nodes.

Proof. Since $T(G)$ is a binary tree, it is sufficient to show that the number of internal nodes is $O(n^2)$. To this end, we prove that the internal nodes of $T(G)$ can be labelled by pairwise different pairs (a, b) , where $a, b \in V(G)$.

Let $G' = (V', E')$ be an induced subgraph of G that corresponds to an internal node X of $T(G)$. If X is a homogeneous node, then G' is decomposed into subgraphs $G_1 = G'[M]$ and $G_2 = G'[(V' \setminus M) \cup \{m\}]$, where $M \subset V'$ is a homogeneous set of G' and m is a vertex in M . In this case, we label X with (a, b) , where $a \in M \setminus \{m\}$ and $b \in V' \setminus M$. If X is an antineighborhood node, then G' is decomposed into subgraphs $G_1 = G' - N(v)$ and $G_2 = G' - v$, where v is a good vertex of G' . In this case, X is labelled with (v, b) , where $b \in N(v)$.

Suppose, to the contrary, that there are two internal nodes A and B in $T(G)$ with the same label (a, b) . By construction, this means that a, b are vertices of both G_A and G_B , the subgraphs of G corresponding to the nodes A and B , respectively. Assume first that B is a descendant of A . The choice of the labels implies that regardless of the type of node A (homogeneous or antineighborhood), the label of A has at least one vertex that is not a vertex of G_B , a contradiction. Now, assume that neither A is a descendant of B nor B is a descendant of A . Let X be the lowest common ancestor of A and B in $T(G)$. If X is a homogeneous node, then G_A and G_B can have at most one vertex in common, and thus A and B cannot have the same label. If X is an antineighborhood node, then one of its children is a leaf, contradicting to the assumption that both A and B are internal nodes. \square

Lemma 4. Let G be an n -vertex graph in \mathcal{G} . If time complexities of the algorithms P and Q are $O(n^p)$ and $O(n^q)$, respectively, then $T(G)$ can be constructed in time $O(n^{2+\max\{2, p, q\}})$.

Proof. The time needed to construct $T(G)$ is the sum of times required to identify types of nodes of $T(G)$ and to decompose graphs corresponding to internal nodes of $T(G)$. To determine the type of a given node X of $T(G)$, we first use the algorithm P to establish whether the graph G_X corresponding to X belongs to \mathcal{F} or not. In the former case X is a leaf node, in the latter case we further try to find in G_X a homogeneous set, which can be performed in $O(n+m)$ time [15]. If G_X has a homogeneous set, then X is a homogeneous node and we decompose G_X into the graphs induced by the vertices in and outside the homogeneous set, respectively. If G_X does not have a homogeneous set, then X is an antineighborhood node,

and the decomposition of G_X is equivalent to finding a good vertex, which can be done by means of the algorithm Q . Since there are $O(n^2)$ nodes in $T(G)$, the total time complexity for constructing $T(G)$ is $O(n^{2+\max\{2,p,q\}})$. \square

Now we are ready to prove the main result of this section.

Theorem 5. *If \mathcal{G} is a hereditary class such that the class \mathcal{G}_p of prime graphs in \mathcal{G} has the antineighborhood property, then the WID problem can be solved in polynomial time for graphs in \mathcal{G} .*

Proof. Let G be an n -vertex graph in \mathcal{G} . To solve the WID problem for G , we construct $T(G)$ and then traverse it bottom-up, deriving a solution for each node of $T(G)$ from the solutions corresponding to the children of that node.

The construction of $T(G)$ requires a polynomial time by Lemma 4. For the instances corresponding to leaf-nodes of $T(G)$, the problem can be solved in polynomial time by the antineighborhood property. According to the discussion in Sections 3.1.1 and 3.1.2, the solution for an instance corresponding to an internal node can be derived from the solutions of its children in polynomial time. Finally, as there are $O(n^2)$ nodes in $T(G)$ (Lemma 3), the total running time to solve the problem for G is polynomial. \square

3.2. Application to $(P_5, \overline{P_5})$ -free graphs

In this section, we show that the WID problem can be solved efficiently for $(P_5, \overline{P_5})$ -free graphs by means of the decomposition scheme described in Section 3.1.3. To this end, we will prove that the class of prime $(P_5, \overline{P_5})$ -free graphs has the antineighborhood property. We start with several auxiliary results. The first of them is simple and we omit its proof.

Observation 2. *Let $G = (V, E)$ be a graph, and let $W \subset V$ induce a connected subgraph in G . If a vertex $v \in V \setminus W$ distinguishes W , then v distinguishes two adjacent vertices of W .*

Proposition 1. *Let $G = (V, E)$ be a prime graph. If a subset $W \subset V$ has at least two vertices and is not a clique, then there exists a vertex $v \in V \setminus W$ which distinguishes two non-adjacent vertices of W .*

Proof. Suppose, to the contrary, that none of the vertices in $V \setminus W$ distinguishes a pair of non-adjacent vertices in W . If $G[W]$ has more than one connected component, then it is easy to see that no vertex outside of W distinguishes W . Hence, W is a homogeneous set in G , which contradicts the primality of G .

If $G[W]$ is connected, then $\overline{G[W]}$ has a connected component C with at least two vertices, since W is not a clique. Then, by our assumption and Observation 2, no vertex outside of W distinguishes C . Also, by the choice of C , no vertex of W outside of C distinguishes C . Therefore, $V(C)$ is a homogeneous set in G . This contradiction completes the proof of the proposition. \square

Lemma 6. *If a $(P_5, \overline{P_5})$ -free prime graph contains an induced copy of $2K_2$, then it has a clique separator.*

Proof. Let $G = (V, E)$ be a $(P_5, \overline{P_5})$ -free prime graph containing an induced copy of $2K_2$. Let $S \subseteq V$ be a minimal separator with the property that $G - S$ contains at least two non-trivial connected components, i.e. connected components with at least two vertices. Such a separator necessarily exists, since G contains an induced $2K_2$.

It follows from the choice of S that

- $G - S$ has $k \geq 2$ connected components C_1, \dots, C_k ;
- $r \geq 2$ of these components, say C_1, \dots, C_r , have at least two vertices, and all the other components C_{r+1}, \dots, C_k are trivial;
- every vertex in S has a neighbour in each of the non-trivial components C_1, \dots, C_r (since S is minimal);
- for every $i \in \{r+1, \dots, k\}$, the unique vertex of the trivial component C_i has a neighbour in S (since G is connected).

In the remaining part of the proof, we show that G has a clique separator. Let us denote $U_i = V(C_i)$ for $i = 1, \dots, k$. We first observe the following.

Claim 1. *Any vertex in S distinguishes at most one of the sets U_1, \dots, U_r .*

Proof. Assume $v \in S$ distinguishes U_i and U_j for distinct $i, j \in [r]$. Then by Observation 2 v distinguishes two adjacent vertices a, b in U_i and two adjacent vertices c, d in U_j . But then a, b, v, c, d induce a forbidden P_5 .

According to Claim 1, the set S can be partitioned into subsets S_0, S_1, \dots, S_r , where the vertices of S_0 dominate every member of $\{U_1, \dots, U_r\}$, and for each $i \in [r]$, the vertices of S_i distinguish U_i and dominate U_j for all j different from i . Moreover, for each $i \in [r]$ the set S_i is non-empty, as the graph G is prime. Now we prove two more auxiliary claims.

Claim 2. For $0 \leq i < j \leq r$, every vertex in S_i is adjacent to every vertex in S_j .

Proof. Assume that the claim is false, i.e. there exist two non-adjacent vertices $s_i \in S_i$ and $s_j \in S_j$. By [Observation 2](#) there exist two adjacent vertices $a, b \in U_j$ that are distinguished by s_j . But then s_i, s_j, a, b and any vertex in $N(s_i) \cap U_i$ induce a forbidden $\overline{P_5}$, a contradiction.

Claim 3. For $i \in [r]$, no vertex in U_i distinguishes two non-adjacent vertices in S_i .

Proof. Assume that there exists a pair of non-adjacent vertices $x, y \in S_i$ that are distinguished by a vertex $u_i \in U_i$. Let $j \in [r] \setminus \{i\}$, and let $s_j \in S_j$ and $u_j \in U_j \setminus N(s_j)$. Then, since s_j dominates S_i , we have that u_j, x, y, s_j, u_i induce a forbidden $\overline{P_5}$, a contradiction.

We split further analysis into two cases.

Case 1: there is at least one trivial component in $G \setminus S$, i.e. $k > r$. For $i \in \{r + 1, \dots, k\}$ we denote by u_i the unique vertex of U_i . Let $U = \{u_{r+1}, \dots, u_k\}$ and let u^* be a vertex in U with a minimal (under inclusion) neighbourhood. We will show that $N(u^*)$ is a clique, and hence is a clique separator in G . By Claim 2, it suffices to show that $N(u^*) \cap S_i$ is a clique for each $i \in \{0, 1, \dots, k\}$. Suppose that for some i the set $N(u^*) \cap S_i$ is not a clique. Then, by [Proposition 1](#), there are two nonadjacent vertices $x, y \in N(u^*) \cap S_i$ distinguished by a vertex $z \in V \setminus (N(u^*) \cap S_i)$. It follows from Claims 2 and 3 that either $z \in S_i \setminus N(u^*)$ or $z \in U$. If $z \in S_i \setminus N(u^*)$, then u^*, x, y, z , and any vertex in $U_j, j \in [r] \setminus \{i\}$ induce a forbidden $\overline{P_5}$, a contradiction. Hence, assume that none of the vertices in $S \setminus (N(u^*) \cap S_i)$ distinguishes two nonadjacent vertices in $N(u^*) \cap S_i$. If $z \in U$, with z being nonadjacent to x and adjacent to y , then by the minimality of $N(u^*)$ there is a vertex $s \in N(z)$ that is not adjacent to u^* . Since $N(z) \subseteq S$, vertex s does not distinguish x and y . But then x, u^*, y, z, s induce either a P_5 (if s is adjacent neither to x nor to y) or a $\overline{P_5}$ (if s is adjacent to both x and y), a contradiction.

Case 2: there are no trivial components in $G \setminus S$, i.e. $k = r$. First, observe that $|S_0| \leq 1$, since G is prime and no vertex outside of S_0 distinguishes S_0 (which follows from the definition of S_0 , Claim 2 and the fact that $k = r$). Further, Claims 2 and 3 imply that for each $i \in [r]$ no vertex in $V \setminus S_i$ distinguishes two nonadjacent vertices in S_i . Therefore, applying [Proposition 1](#) we conclude that S_i is a clique. Hence $S = \bigcup_{i=0}^r S_i$ is a clique separator in G . \square

Lemma 7. Let G be a $(P_5, \overline{P_5})$ -free prime graph containing an induced copy of $2K_2$. Then G contains a permissible antisimplicial vertex.

Proof. By [Lemma 6](#) graph G has a clique separator, and therefore it also has a minimal clique separator S . Let $C_1, \dots, C_k, k \geq 2$, be connected components of $G - S$, and $U_i = V(C_i), i = 1, \dots, k$. Since S is a minimal separator, every vertex in S has at least one neighbour in each of the sets U_1, \dots, U_k . By Claim 1 in the proof of [Lemma 6](#), any vertex in S distinguishes at most one of the sets U_1, \dots, U_k , and therefore, the set S partitions into subsets S_0, S_1, \dots, S_k , where the vertices of S_0 dominate every member of $\{U_1, \dots, U_k\}$, and for each $i \in [k]$ the vertices of S_i distinguish U_i and dominate U_j for all j different from i .

If $S_0 \neq \emptyset$, then any vertex in S_0 is adjacent to all the other vertices in the graph, and therefore it is permissible and antisimplicial. Hence, without loss of generality, assume that $S_0 = \emptyset$ and $S_1 \neq \emptyset$.

Let s be a vertex in S_1 with a maximal (under inclusion) neighbourhood in U_1 . We will show that s is antisimplicial and permissible. Suppose that the graph induced by the antineighbourhood of s contains a connected component C with at least two vertices. Since G is prime, by [Observation 2](#) it must contain a vertex p outside of C distinguishing two adjacent vertices q and t in C . Then p does not belong to $N(s) \cap U_1$, since otherwise q, t, p, s together with any vertex in U_2 would induce a P_5 . Therefore, p belongs to S_1 . Since the set $N(s) \cap U_1$ is maximal, it contains a vertex y nonadjacent to p . But now t, q, p, s, y induce either a P_5 or its complement, as y does not distinguish q and t . This contradiction shows that every component in the graph induced by the antineighbourhood of s is trivial, i.e. s is antisimplicial.

Assume now that s is not permissible, i.e. there exists an independent dominating set I in $G - s$ that does not contain a neighbour of s . Since s dominates $U_2 \cup \dots \cup U_k$, the set I is a subset of $U_1 \setminus N(s)$. But then I is not dominating, since no vertex of U_2 has a neighbour in I . This contradiction completes the proof of the lemma. \square

Lemma 8. The class of prime $(P_5, \overline{P_5})$ -free graphs has the antineighborhood property.

Proof. Let \mathcal{F} be the class of $(2K_2, \overline{P_5})$ -free graphs (this is a subclass of $(P_5, \overline{P_5})$ -free graphs, since $2K_2$ is an induced subgraph of P_5). Clearly, graphs in \mathcal{F} can be recognised in polynomial time. Moreover, the WID problem can be solved in polynomial time for graphs in \mathcal{F} , because the problem is polynomially solvable on $2K_2$ -free graphs (according to [\[1\]](#), these graphs have polynomially many maximal independent sets).

If a prime $(P_5, \overline{P_5})$ -free graph $G = (V, E)$ does not belong to \mathcal{F} , then by [Lemma 7](#) it contains a permissible vertex v whose antineighbourhood is an independent set, and therefore, $G - N(v) \in \mathcal{F}$. It remains to check that a permissible

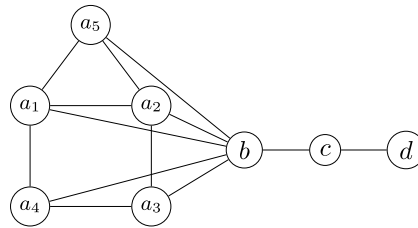


Fig. 3. The graph $\overline{P_5}^*$.

antisimplicial vertex in G can be found in polynomial time. It follows from the proof of Lemma 7 that in a minimal clique separator of G any vertex with a maximal neighbourhood is permissible and antisimplicial. A minimal clique separator in a graph can be found in polynomial time [17], and therefore the desired vertex can also be computed efficiently. \square

Now the main result of the section follows from Theorem 5 and Lemma 8.

Theorem 9. *The WID problem is polynomial-time solvable in the class of $(P_5, \overline{P_5})$ -free graphs.*

4. WID in $(P_5, \overline{P_3 + P_2})$ -free graphs

To solve the problem for $(P_5, \overline{P_3 + P_2})$ -free graphs, let us introduce the following notation: for an arbitrary graph F , we denote by F^* the graph obtained from F by adding three new vertices, say b, c, d , such that b dominates (adjacent to each vertex of) F , while c is adjacent to b and d only (see Fig. 3 for an illustration in the case $F = \overline{P_5}$). The importance of this notation is due to the following result proved in [13].

Theorem 10. *Let F be any connected graph. If the WID problem can be solved in polynomial time for (P_5, F) -free graphs, then this problem can also be solved in polynomial time for (P_5, F^*) -free graphs.*

This result together with Theorem 9 leads to the following conclusion.

Corollary 1. *The WID problem is polynomial-time solvable in the class of $(P_5, \overline{P_5}^*)$ -free graphs.*

To solve the problem for $(P_5, \overline{P_3 + P_2})$ -free graphs, in this section we reduce it to $(P_5, \overline{P_3 + P_2}, \overline{P_5}^*)$ -free graphs, where the problem is solvable in polynomial time by Corollary 1.

Let G be a $(P_5, \overline{P_3 + P_2})$ -free graph containing a copy of $\overline{P_5}^*$ induced by vertices $a_1, a_2, a_3, a_4, a_5, b, c, d$, as shown in Fig. 3.

Denote by U the set of vertices in G that have at least one neighbour in $\{a_1, a_2, a_3, a_4, a_5\}$, that is, $U = N(a_1) \cup \dots \cup N(a_5)$. In particular, $\{a_1, a_2, a_3, a_4, a_5, b\}$ is a subset of U . We assume that

(**) the copy of $\overline{P_5}^*$ in G is chosen in such a way that U has the minimum number of elements.

Now we prove several auxiliary results about the structure of G .

Proposition 2. *If a vertex $x \in U$ has a neighbour y outside of U , then x is adjacent to each of the vertices a_1, a_2, a_3, a_4 .*

Proof. Let $A = \{a_1, a_2, a_3, a_4\}$. Note that if x is adjacent to a_5 , then it must be adjacent to at least one vertex in A , since otherwise a forbidden P_5 arises. If x is adjacent to exactly one or to exactly two adjacent vertices in A , then $\{x, y\} \cup A$ induces a subgraph containing a forbidden P_5 . If x is adjacent to exactly two non-adjacent vertices in A , say a_1 and a_3 , then x must be adjacent to a_5 , since otherwise y, x, a_3, a_2, a_5 induce a P_5 . But this is impossible, since in this case x, a_1, a_2, a_3, a_5 induce a $\overline{P_3 + P_2}$. Finally, if x has exactly three neighbours in A , then $\{x\} \cup A$ induces a forbidden $\overline{P_3 + P_2}$. Therefore, x must be adjacent to every vertex in A . \square

Taking into account Proposition 2, we partition the set U into three subsets as follows:

U_1 consists of the vertices of U that are adjacent to each of the vertices a_1, a_2, a_3, a_4 , and have at least one neighbour outside of U ;

U_2 consists of the vertices of U that are adjacent to each of the vertices a_1, a_2, a_3, a_4 , but have no neighbours outside of U ;

$U_3 = U \setminus (U_1 \cup U_2)$.

Notice that U_1 is non-empty as it contains b . Also $\{a_1, a_2, a_3, a_4, a_5\} \subseteq U_3$, and no vertex in U_3 has a neighbour outside of U .

Proposition 3. U_1 is a clique in G .

Proof. Suppose to the contrary that U_1 contains two non-adjacent vertices x_1 and x_2 . Also, let y_1 and y_2 be neighbours of x_1 and x_2 outside of U , respectively. Vertex y_1 is not adjacent to x_2 , since otherwise x_1, x_2, a_1, a_2, y_1 induce a $\overline{P_3 + P_2}$. Similarly, y_2 is not adjacent to x_1 . Hence $y_1 \neq y_2$, and therefore, to avoid a copy of P_5 induced by y_1, x_1, a_1, x_2, y_2 , vertices y_1 and y_2 must be adjacent. For the same reason, a_5 should be adjacent to both x_1 and x_2 . But then x_1, x_2, a_3, a_4, a_5 induce a copy of the forbidden $\overline{P_3 + P_2}$, a contradiction. \square

Proposition 4. The graph $G[U_2 \cup U_3]$ is $\overline{P_5^*}$ -free.

Proof. Suppose to the contrary that $G[U_2 \cup U_3]$ contains vertices $a'_1, a'_2, a'_3, a'_4, a'_5, b', c', d'$ inducing a $\overline{P_5^*}$ (similarly to Fig. 3). Since no vertex in $U_2 \cup U_3$ has a neighbour outside of U in G , and c', d' are not adjacent to any of the vertices $a'_1, a'_2, a'_3, a'_4, a'_5$, we conclude that $|N(a'_1) \cup \dots \cup N(a'_5)| \leq |U| - 2$, which contradicts the minimality of $|U|$. \square

Now we describe a reduction from the graph G with a weight function w to a graph G' with a weight function w' , where $|V(G')| \leq |V(G)| - 4$, G' is $(P_5, \overline{P_3 + P_2})$ -free, and $id_w(G) = id_{w'}(G')$. First, we define G' as the graph obtained from G by

1. removing the vertices of U_3 ;
2. adding edges between any two non-adjacent vertices in $U_1 \cup U_2$;
3. adding a new vertex u adjacent to every vertex in $U_1 \cup U_2$.

Clearly, $|V(G')| \leq |V(G)| - 4$, as the set U_3 of the removed vertices contains at least 5 elements and we add exactly one new vertex u . In the next proposition, we show that the above reduction does not produce any of the forbidden subgraphs.

Proposition 5. The graph G' is $(P_5, \overline{P_3 + P_2})$ -free.

Proof. Note that the graph $G' - (U_2 \cup \{u\})$ is isomorphic to $G - (U_2 \cup U_3)$, and therefore it contains no P_5 or $\overline{P_3 + P_2}$ as an induced subgraph. Hence, if G' contains a forbidden subgraph, then at least one of the vertices of this subgraph should lie in $U_2 \cup \{u\}$.

By construction of G' and the definition of U_2 , the set $U_2 \cup \{u\}$ is a clique, and every vertex in this set is simplicial in G' . Therefore, no vertex of $U_2 \cup \{u\}$ can be a part of an induced copy of $\overline{P_3 + P_2}$. Also, $U_2 \cup \{u\}$ can contain at most one vertex of an induced copy of P_5 , and if $U_2 \cup \{u\}$ contains such a vertex, it must be a degree-one vertex of the P_5 . Suppose to the contrary that G' contains a copy of P_5 induced by v_1, v_2, v_3, v_4, v_5 with $v_1 \in U_2 \cup \{u\}$ and $\{v_2, v_3, v_4, v_5\} \subseteq V(G') \setminus (U_2 \cup \{u\})$. But then a_1, v_2, v_3, v_4, v_5 induce a forbidden P_5 in G , a contradiction. \square

Now we define a weight function w' on the vertex set of G' as follows:

1. $w'(x) = w(x)$, for every $x \in V(G') \setminus (\{u\} \cup U_1 \cup U_2)$;
2. $w'(u) = id_w(G[U_3])$;
3. $w'(x) = w(x) + id_w(G[U \setminus N[x]])$, for every $x \in U_1$;
4. $w'(x) = w(x) + id_w(G[U \setminus (U_1 \cup N[x])])$, for every $x \in U_2$.

Lemma 11. Given a weighted graph (G, w) , the weighted graph (G', w') can be constructed in polynomial time.

Proof. To construct G' we need to find in G an induced copy of $\overline{P_5^*}$ that minimises $|U|$. Clearly, this can be done in polynomial time.

To show that w' can be computed in polynomial time we observe that each of the graphs $G[U_3]$, $G[U \setminus (U_1 \cup N[x])]$ for $x \in U_2$, and $G[U \setminus N[x]]$ for $x \in U_1$ is an induced subgraph of $G[U_2 \cup U_3]$. This observation together with Proposition 4 and Corollary 1 imply the desired conclusion and finish the proof of the lemma. \square

Now let us show that $id_w(G) = id_{w'}(G)$. For this, we will need two auxiliary propositions.

Proposition 6. Any independent dominating set in $G[U_3]$ dominates $U_1 \cup U_2$.

Proof. Let $A = \{a_1, a_2, a_3, a_4\}$, and let I be an independent dominating set in $G[U_3]$. If I contains at least one of the vertices from A , then I dominates $U_1 \cup U_2$, so we assume that $I \subseteq U_3 \setminus A$. Note that a vertex $x \in U_3 \setminus A$ has at most two neighbours in A . Indeed, x cannot have four neighbours by the definition of U_3 , and it cannot have three neighbours, since otherwise $\{x\} \cup A$ induces a forbidden $\overline{P_3 + P_2}$. Now, if I contains a vertex $x \in U_3 \setminus A$ that is adjacent to a_1 and a_3 , then I dominates $U_1 \cup U_2$, since otherwise x together with a_1, a_2, a_3 and a non-neighbour of x in $U_1 \cup U_2$ induce a forbidden $\overline{P_3 + P_2}$.

Assume that I contains none of the above vertices. Then there exist vertices $x, y \in I$ such that x is adjacent to a_1 and non-adjacent to a_3 , and y is adjacent to a_3 and non-adjacent to a_1 . If I does not dominate $U_1 \cup U_2$, then there exists a vertex $z \in U_1 \cup U_2$ that is adjacent neither to x nor to y . But then x, a_1, z, a_3, y induce a forbidden P_5 . \square

Proposition 7. For every vertex $x \in U_2$, any independent dominating set in the graph $G - U$ dominates $U_1 \setminus N(x)$.

Proof. Suppose to the contrary that there exists an independent dominating set I in the graph $G - U$ that does not dominate a vertex $y \in U_1 \setminus N(x)$. By the definition of U_1 , vertex y has a neighbour z in $V(G) \setminus U$. Since I is dominating in $G - U$, there exists a vertex $v \in I$ that is adjacent to z . But then v, z, y, a_1, x induce a forbidden P_5 , a contradiction. \square

Lemma 12. For any weighted graph (G, w) , we have $id_w(G) = id_{w'}(G')$.

Proof. First, we show that $id_w(G) \geq id_{w'}(G')$. Let I be an independent dominating set of the minimum weight in G . We distinguish between the following three cases:

1. $I \cap U_1 \neq \emptyset$.

By Propositions 2 and 3, the set U_1 is a clique separating $V(G) \setminus U$ from $U \setminus U_1$. Therefore, I has only one element in U_1 , say x , and:

$$id_w(G) = w(x) + id_w(G[U \setminus N[x]]) + id_w(G - (U \cup N[x])).$$

Consequently

$$id_w(G) = w'(x) + id_{w'}(G' - N[x]) \geq id_{w'}(G').$$

2. $I \cap U_1 = \emptyset$ and $I \cap U_2 \neq \emptyset$.

Let $x \in I \cap U_2$. Then using Proposition 7

$$id_w(G) = w(x) + id_w(G[U \setminus (U_1 \cup N[x])]) + id_w(G - U) = w'(x) + id_{w'}(G' - N[x]) \geq id_{w'}(G').$$

3. $I \cap (U_2 \cup U_1) = \emptyset$.

In this case, taking into account Proposition 6, we conclude that

$$id_w(G) = id_w(G[U_3]) + id_w(G - U) = w'(u) + id_{w'}(G' - N[u]) \geq id_{w'}(G').$$

Let us now prove the reverse inequality $id_w(G) \leq id_{w'}(G')$. Let I be an independent dominating set of the minimum weight in G' . Since u does not have neighbours outside of $U_1 \cup U_2$, and $\{u\} \cup U_1 \cup U_2$ is a clique in G' , the set I has exactly one element in $\{u\} \cup U_1 \cup U_2$, which we denote by x . Similarly to the first part of the proof, we consider three cases:

1. $x \in U_1$.

In this case

$$id_{w'}(G') = w'(x) + id_{w'}(G' - N[x]) = w(x) + id_w(G[U \setminus N[x]]) + id_w(G - (U \cup N[x])) \geq id_w(G).$$

2. $x \in U_2$.

In this case, by Proposition 7,

$$id_{w'}(G') = w'(x) + id_{w'}(G' - N[x]) = w(x) + id_w(G[U \setminus (U_1 \cup N[x])]) + id_w(G - U) \geq id_w(G).$$

3. $x = u$.

In this case, by Proposition 6,

$$id_{w'}(G') = w'(x) + id_{w'}(G' - N[x]) = id_w(G[U_3]) + id_w(G - U) \geq id_w(G). \quad \square$$

Now we are ready to prove the main result of this section.

Theorem 13. The WID problem is solvable in polynomial time for $(P_5, \overline{P_3 + P_2})$ -free graphs.

Proof. Let (G, w) be an n -vertex $(P_5, \overline{P_3 + P_2})$ -free weighted graph. If G contains an induced copy of $\overline{P_5}^*$, then by Proposition 5, and Lemmas 11 and 12, the graph (G, w) can be transformed in polynomial time into a $(P_5, \overline{P_3 + P_2})$ -free weighted graph (G', w') with at most $n - 4$ vertices such that $id_w(G) = id_{w'}(G')$. Repeating this procedure at most $\lfloor n/4 \rfloor$ times we obtain a $(P_5, \overline{P_3 + P_2}, \overline{P_5}^*)$ -free weighted graph (H, σ) such that $id_w(G) = id_\sigma(H)$. By Corollary 1 the WID problem for (H, σ) can be solved in polynomial time. Finally, it is not difficult to see that a polynomial-time procedure computing $id_w(G)$ can be easily transformed into a polynomial-time algorithm finding an independent dominating set of weight $id_w(G)$. \square

5. Concluding remarks and open problems

In this paper, we proved that WEIGHTED INDEPENDENT DOMINATION can be solved in polynomial time for $(P_5, \overline{P_5})$ -free graphs and $(P_5, \overline{P_3 + P_2})$ -free graphs. A natural question to ask is whether these results can be extended to a class defined by one forbidden induced subgraph.

From the results in [3] it follows that in the case of one forbidden induced subgraph H the problem is solvable in polynomial time *only if* H is a linear forest, i.e. a graph every connected component of which is a path. On the other hand, it is known that this necessary condition is not sufficient, since INDEPENDENT DOMINATION is NP-hard in the class of $2P_3$ -free graphs. This follows from the fact that all sat-graphs are $2P_3$ -free [19].

In the case of a *disconnected* forbidden graph H , polynomial-time algorithms to solve WEIGHTED INDEPENDENT DOMINATION are known only for mP_2 -free graphs for any fixed value of m . This follows from a polynomial bound on the number of maximal independent sets in these graphs [1]. The unweighted version of the problem can also be solved for $P_2 + P_3$ -free graphs [13]. However, for weighted graphs in this class the complexity status of the problem is unknown.

Open Problem 1. Determine the complexity status of WEIGHTED INDEPENDENT DOMINATION in the class of $P_2 + P_3$ -free graphs.

In the case of a *connected* forbidden graph H , i.e. in the case when $H = P_k$, the complexity status is known for $k \geq 7$ (as P_7 contains a $2P_3$) and for $k \leq 4$ (as P_4 -free graphs are precisely the cographs). Therefore, the only open cases are P_5 -free and P_6 -free graphs. As we mentioned in the introduction, the related problem of finding a maximum weight independent set (WIS) has been recently solved for P_5 -free graphs [12]. This result makes the class of P_5 -free graphs of particular interest for WEIGHTED INDEPENDENT DOMINATION and we formally state it as an open problem.

Open Problem 2. Determine the complexity status of WEIGHTED INDEPENDENT DOMINATION in the class of P_5 -free graphs.

We also mentioned earlier that a polynomial-time solution for WIS in a hereditary class \mathcal{X} does not necessarily imply the same conclusion for WID in \mathcal{X} . However, in the reverse direction such examples are not known. We believe that such examples do not exist and propose this idea as a conjecture.

Conjecture 1. If WID admits a polynomial-time solution in a hereditary class \mathcal{X} , then so does WIS.

Acknowledgements

The results of Sections 2 and 3 were obtained under financial support of the Russian Science Foundation grant No. 17-11-01336. The results of Section 4 were obtained under financial support of the Russian Foundation for Basic Research, grant No. 16-31-60008-mol-a-dk, RF President grant MK-4819.2016.1 and LATNA laboratory, National Research University Higher School of Economics.

References

- [1] E. Balas, C.S. Yu, On graphs with polynomially solvable maximum-weight clique problem, *Networks* 19 (1989) 247–253.
- [2] H.L. Bodlaender, A. Brandstädt, D. Kratsch, M. Rao, J. Spinrad, On algorithms for (P_5, gem) -free graphs, *Theoret. Comput. Sci.* 349 (2005) 2–21.
- [3] R. Boliac, V. Lozin, Independent domination in finitely defined classes of graphs, *Theoret. Comput. Sci.* 301 (2003) 271–284.
- [4] B.-M. Bui-Xuan, J.A. Telle, M. Vatshelle, Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems, *Theoret. Comput. Sci.* 511 (2013) 66–76.
- [5] G.J. Chang, The weighted independent domination problem is NP-complete for chordal graphs, *Discrete Appl. Math.* 143 (2004) 351–352.
- [6] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory Comput. Syst.* 33 (2000) 125–150.
- [7] P. Damaschke, H. Muller, D. Kratsch, Domination in convex and chordal bipartite graphs, *Inform. Process. Lett.* 36 (1990) 231–236.
- [8] M. Farber, Independent domination in chordal graphs, *Oper. Res. Lett.* 1 (1982) 134–138.
- [9] V. Giakoumakis, I. Rusu, Weighted parameters in $(P_5, \overline{P_5})$ -free graphs, *Discrete Appl. Math.* 80 (1997) 255–261.
- [10] D.V. Korobitsyn, On the complexity of determining the domination number in monogenic classes of graphs, (*Russian*) *Diskret. Mat.* 2 (3) (1990) 90–96; translation in *Discrete Math. Appl.* 2 (2) (1992) 191–199.
- [11] T. Karthick, On atomic structure of P_5 -free subclasses and maximum weight independent set problem, *Theoret. Comput. Sci.* 516 (2014) 78–85.
- [12] D. Lokshantov, M. Vatshelle, Y. Villanger, Independent set in P_5 -free graphs in polynomial time, in: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, ACM, New York, 2014, pp. 570–581.

- [13] V. Lozin, R. Mosca, C. Purcell, Independent domination in finitely defined classes of graphs: polynomial algorithms, *Discrete Appl. Math.* 182 (2015) 2–14.
- [14] V. Lozin, D. Malyshev, R. Mosca, V. Zamaraev, New results on weighted independent domination, in: *Lecture Notes in Computer Science*, 2017, to appear.
- [15] R.M. McConnell, J. Spinrad, Modular decomposition and transitive orientation, *Discrete Math.* 201 (1999) 189–241.
- [16] R.H. Möhring, F.J. Radermacher, Substitution decomposition for discrete structures and connections with combinatorial optimization, *Ann. Discrete Math.* 19 (1984) 257–356.
- [17] S.H. Whitesides, An algorithm for finding clique cut-sets, *Inform. Process. Lett.* 12 (1981) 31–32.
- [18] M. Yannakakis, F. Gavril, Edge dominating sets in graphs, *SIAM J. Appl. Math.* 38 (1980) 364–372.
- [19] I.E. Zverovich, Satgraphs and independent domination. Part 1, *Theoret. Comput. Sci.* 352 (2006) 47–56.