New Results on Weighted Independent Domination

Vadim Lozin[1(✉)], Dmitriy Malyshev[2], Raffaele Mosca[3], and Viktor Zamaraev[1]

[1] Mathematics Institute, University of Warwick, Coventry CV4 7AL, UK
V.Lozin@warwick.ac.uk
[2] National Research University Higher School of Economics,
25/12 Bolshaya Pecherskaya Ulitsa, 603155 Nizhny Novgorod, Russia
[3] Dipartimento di Economia, Universitá degli Studi "G. D'Annunzio",
65121 Pescara, Italy

Abstract. Weighted independent domination is an NP-hard graph problem, which remains computationally intractable in many restricted graph classes. Only few examples of classes are available, where the problem admits polynomial-time solutions. In the present paper, we extend the short list of such classes with two new examples.

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 optimization 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 [15]. It is also NP-hard for planar graphs, triangle-free graphs, graphs of vertex degree at most 3 [3], line graphs [14], chordal bipartite graphs [5], 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 [4], where independent domination can be solved in polynomial time [6].

Not much is known about graph classes allowing an efficient solution of the WID problem. Among rare examples of this type, let us mention cographs and split graphs.

- 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. 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. 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.

H.L. Bodlaender and G.J. Woeginger (Eds.): WG 2017, LNCS 10520, pp. 399–411, 2017.
https://doi.org/10.1007/978-3-319-68705-6_30

Let us observe that in both these examples we deal with *hereditary classes*, i.e. with classes of graphs 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 characterized in terms of minimal forbidden induced subgraphs. For instance, the cographs are precisely P_4-free graphs (i.e. graphs containing no induced P_4), while the split graphs are the graphs which are free of $2K_2, C_4$ and C_5.

The class of sat-graphs (as well as each of the other classes mentioned earlier) also is hereditary. It 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 more extensions of split graphs: the class of $(P_5, \overline{P}_5)$-free graphs and the class of $(P_5, \overline{P_3 + P_2})$-free graphs. The first of them also extends the cographs, since both forbidden graphs contain a P_4. From an algorithmic point of view, both extensions are 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. For the second class, we develop a tricky reduction allowing us to reduce the problem to the first class.

Let us emphasize 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. 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 [9] after several decades of attacking the problem on subclasses of P_5-free graphs (see e.g. [2,7,8]). 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. Determining the complexity status of WID in P_5-free graphs 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 organized as follows: Sect. 2 contains preliminary information, in Sect. 3 we solve the problem for $(P_5, \overline{P}_5)$-free graphs, and in Sect. 4 we solve it for $(P_5, \overline{P_3 + P_2})$-free graphs.

2 Preliminaries

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 $\{1, 2, \ldots, n\}$ we will often use the notation $[n]$.

2.1 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. [12] 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

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

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.

2.2 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.

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

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 1. *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 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 Sect. 3.1 that combines modular decomposition and antineighborhood decomposition. In Sect. 3.2 we apply it to $(P_5, \overline{P}_5)$-free graphs.

3.1 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 1. *For an n-vertex graph $G \in \mathcal{G}$, 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 labeled 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 labeled 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. □

Lemma 2. *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 [11]. 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 $O(n^{2+\max\{2,p,q\}})$. □

Theorem 2. *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 2. 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 Sects. 2.1 and 2.2, 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 1), the total running time to solve the problem for G is polynomial. □

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 Sect. 3.1. 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 1. *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 1, 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. □

Lemma 3. *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 1 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 1 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 $i \in \{r+1, \ldots, k\}$ we denote by u_i the unique vertex of U_i. Let $U = \{u_{r+1}, \ldots, u_k\}$ 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 $i \in \{0, 1, \ldots, k\}$. 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. □

Lemma 4. *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 3, G has a clique separator, and therefore it also has a minimal clique separator S. Let $C_1, \ldots, C_k$, $k \geq 2$, be connected components of $G - S$, and $U_i = V(C_i)$, $i = 1, \ldots, k$. Since S is a minimal separator, every vertex in S has at least one neighbour in each of the sets $U_1, \ldots, U_k$. By Claim 1 in the proof of Lemma 3, any vertex in S distinguishes at most one of the sets $U_1, \ldots, U_k$, and therefore, the set S partitions into subsets $S_0, S_1 \ldots, S_k$, where the vertices of S_0 dominate every member of $\{U_1, \ldots, U_k\}$, 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 1 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 $U_2 \cup \ldots \cup U_k$, 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. □

Lemma 5. *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 recognized in polynomial time. 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 4 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 antisimplicial vertex in G can be found in polynomial time. It follows from the proof of Lemma 4 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 [13], and therefore the desired vertex can also be computed efficiently. □

Now the main result of the section follows from Theorem 2 and Lemma 5.

Theorem 3. *The WID problem can be solved in polynomial time 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, we introduce the following notation: for an arbitrary graph F, let F^* be the graph obtained from F by adding three new vertices, say b, c, d, such that b is adjacent to each vertex of F, while c is adjacent to b and d only (see Fig. 1 for an illustration in the case $F = \overline{P}_5$). The importance of this notation is due to the following result.

Theorem 4 [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 3 leads to the following conclusion.

Corollary 1. *The WID problem can be solved in polynomial time 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 what follows we reduce it to $(P_5, \overline{P_3 + P_2}, \overline{P_5}^*)$-free graphs, where the problem is solvable 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. 1.

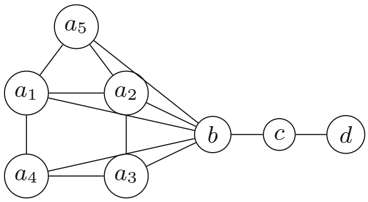

Fig. 1. The graph $\overline{P_5}^*$

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, $U = N(a_1) \cup \ldots \cup N(a_5)$. In particular, U includes $\{a_1, a_2, a_3, a_4, a_5, b\}$. We assume that

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

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.*

Proposition 2 allows us to 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.*

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

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.*

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

To show that $id_w(G) = id'_w(G)$, we need two auxiliary propositions.

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

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

Lemma 7. *For any weighted graph (G, w), we have $id_w(G) = id_{w'}(G')$.*

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

Theorem 5. *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 6 and 7, 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. To conclude the proof we observe 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 $id_w(G)$. □

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 [15].

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 [10]. However, for weighted graphs in this class the complexity status of the problem is unknown.

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 [9]. 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.

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 Sect. 3 were obtained under financial support of the Russian Science Foundation grant No. 17-11-01336. The results of Sect. 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. Balas, E., Yu, C.S.: On graphs with polynomially solvable maximum-weight clique problem. Networks **19**, 247–253 (1989)
2. Bodlaender, H.L., Brandstädt, A., Kratsch, D., Rao, M., Spinrad, J.: On algorithms for (P_5, gem)-free graphs. Theoret. Comput. Sci. **349**, 2–21 (2005)
3. Boliac, R., Lozin, V.: Independent domination in finitely defined classes of graphs. Theoret. Comput. Sci. **301**, 271–284 (2003)
4. Chang, G.J.: The weighted independent domination problem is NP-complete for chordal graphs. Discret. Appl. Math. **143**, 351–352 (2004)
5. Damaschke, P., Muller, H., Kratsch, D.: Domination in convex and chordal bipartite graphs. Inf. Process. Lett. **36**, 231–236 (1990)
6. Farber, M.: Independent domination in chordal graphs. Oper. Res. Lett. **1**, 134–138 (1982)
7. Giakoumakis, V., Rusu, I.: Weighted parameters in $(P_5, \overline{P}_5)$-free graphs. Discret. Appl. Math. **80**, 255–261 (1997)
8. Karthick, T.: On atomic structure of P_5-free subclasses and maximum weight independent set problem. Theoret. Comput. Sci. **516**, 78–85 (2014)
9. Lokshantov, D., Vatshelle, M., Villanger, Y.: Independent set in P_5-free graphs in polynomial time. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 570–581 (2014)
10. Lozin, V., Mosca, R., Purcell, C.: Independent domination in finitely defined classes of graphs: polynomial algorithms. Discret. Appl. Math. **182**, 2–14 (2015)
11. McConnell, R.M., Spinrad, J.: Modular decomposition and transitive orientation. Discret. Math. **201**, 189–241 (1999)
12. Möhring, R.H., Radermacher, F.J.: Substitution decomposition for discrete structures and connections with combinatorial optimization. Ann. Discret. Math. **19**, 257–356 (1984)
13. Whitesides, S.H.: An algorithm for finding clique cut-sets. Inf. Process. Lett. **12**, 31–32 (1981)
14. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. SIAM J. Appl. Math. **38**, 364–372 (1980)
15. Zverovich, I.E.: Satgraphs and independent domination. Part 1. Theoret. Comput. Sci. **352**, 47–56 (2006)