



New Results on Independent Sets in Extensions of $2K_2$ -free Graphs

Raffaele Mosca¹

Received: 1 May 2021 / Accepted: 7 July 2022 / Published online: 29 July 2022
© The Author(s) 2022

Abstract

The Maximum Weight Independent Set Problem (WIS) is a well-known NP-hard problem. A popular way to study WIS is to detect graph classes for which WIS is solvable in polynomial time, with particular reference to hereditary graph classes, i. e., defined by a hereditary graph property or equivalently by forbidding one or more induced subgraphs. For any two graphs G and H , $G + H$ denotes the disjoint union of G and H . A lK_2 is the disjoint union of l edges. A $Y_{m,m}$ is the disjoint union of two stars of $m + 1$ vertices plus one vertex that is adjacent only to the centers of such stars. For any graph family \mathcal{Y} , the class of \mathcal{Y} -free graphs is formed by graphs which are Y -free for every $Y \in \mathcal{Y}$, and the class of $lK_2 + \mathcal{Y}$ -free graphs is formed by graph which are $lK_2 + Y$ -free for every $Y \in \mathcal{Y}$. The main result of this manuscript is the following: For any constant m and for any graph family \mathcal{Y} which contains an induced subgraph of $Y_{m,m}$, if WIS is solvable in polynomial time for \mathcal{Y} -free graphs, then WIS is solvable in polynomial time for $lK_2 + \mathcal{Y}$ -free graphs for any constant l . That extends some known polynomial results, namely, when $\mathcal{Y} = \{Y\}$ and Y is a fork or is a P_5 . The proof of the main result is based on Farber's approach to prove that every $2K_2$ -free graph has $O(n^2)$ maximal independent sets (Farber in Discrete Math 73:249–260, 1989), which directly leads to a polynomial time algorithm to solve WIS for $2K_2$ -free graphs through a dynamic programming approach, and on some extensions of Farber's approach.

Keywords Maximum independent set problem · Polynomial algorithms · $2K_2$ -free graphs · Hereditary graph classes

1 Introduction

For any missing notation or reference let us refer to [6].

✉ Raffaele Mosca
r.mosca@unich.it

¹ Dipartimento di Economia, Università degli Studi "G. d'Annunzio", 65127 Pescara, Italy

For any graph G , let $V(G)$ and $E(G)$ denote respectively the vertex-set and the edge-set of G . Let G be a graph. For any vertex-set $U \subseteq V(G)$, let $N_G(U) = \{v \in V(G) \setminus U : v \text{ is adjacent to some } u \in U\}$ be the *neighborhood of U in G* , and $A_G(U) = V(G) \setminus (U \cup N(U))$ be the *anti-neighborhood of U in G* . If $U = \{u_1, \dots, u_k\}$, then let us simply write $N_G(u_1, \dots, u_k)$ instead of $N_G(U)$, and $A_G(u_1, \dots, u_k)$ instead of $A_G(U)$. For any subset $U \subseteq V(G)$ let $G[U]$ be the subgraph of G induced by U . For any vertex $v \in V(G)$ and for any subset $U \subset V(G)$, with $v \notin U$, let us say that: v *contacts* U if v is adjacent to some vertex of U ; v *dominates* U if v is adjacent to each vertex of U . For any vertex-sets $U, W \subseteq V(G)$, let us say that U *has a co-join* to W (or generally that U and W *have mutually a co-join*) if $U \cap W = \emptyset$ and if each vertex of U is non-adjacent to each vertex of W . A *component of G* is the vertex set of a maximal connected subgraph of G . A component of G is *trivial* if it is a singleton, and *nontrivial* otherwise. A *clique* of G is a set of pairwise adjacent vertices of G . An *independent set* (or a *stable set*) of a graph G is a subset of pairwise nonadjacent vertices of G . An independent set of G is *maximal* if it is not properly contained in another independent set of G .

The following specific graphs are mentioned later. A P_k has vertices v_1, v_2, \dots, v_k and edges $v_j v_{j+1}$ for $1 \leq j < k$. A C_k has vertices v_1, v_2, \dots, v_k and edges $v_j v_{j+1}$ for $1 \leq j < k$ and $v_k v_1$. A K_n is a complete graph of n vertices. A $K_{1,n}$ —which is also called a *star of $n + 1$ vertices*—is a complete bipartite graph whose sides respectively have one vertex, called the *center* of the star, and n vertices, called the *leaves* of the star (if $n = 1$ then there are two trivial centers). A *claw* has vertices a, b, c, d , and edges ab, ac, ad ; then a claw is a $K_{1,3}$. A *fork* has vertices a, b, c, d, e , and edges ab, ac, ad, de ; then a fork contains both a P_4 and a claw as induced subgraphs. A $Y_{m',m''}$, for some constants m', m'' , is the graph formed by two disjoint stars $K_{1,m'}$ and $K_{1,m''}$ plus one vertex which is adjacent only to the centers of such stars; a $Y_{m,m}$, for some constant m , is depicted in Fig. 1.

A graph G is *H-free*, for a given graph H , if G contains no induced subgraph isomorphic to H ; in particular, H is called a forbidden induced subgraph of G . Given two graphs G and H , $G + H$ denotes the disjoint union of G and H ; in particular, $lG = G + G + \dots + G$ denotes the disjoint union of l copies of G . For any graph family \mathcal{Y} , the class of \mathcal{Y} -free graphs is formed by graphs which are Y -free for every

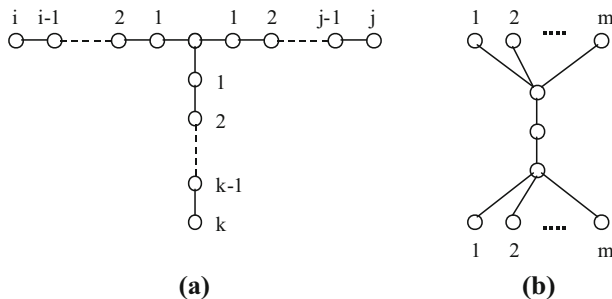


Fig. 1 a $S_{i,j,k}$; b $Y_{m,m}$

$Y \in \mathcal{Y}$, and the class of $lK_2 + \mathcal{Y}$ -free graphs is formed by graph which are $lK_2 + Y$ -free for every $Y \in \mathcal{Y}$.

The *Maximum Weight Independent Set Problem (WIS)* is the following: Given a graph G and a weight function w on $V(G)$, determine an independent set of G of maximum weight, where the weight of an independent set I is given by the sum of $w(v)$ for $v \in I$. Let $\alpha_w(G)$ denote the maximum weight of any independent set of G . WIS reduces to the *Maximum Independent Set Problem (MIS)* if all vertices v have the same weight $w(v) = 1$.

WIS is NP-hard [19, 23]. It remains NP-hard under various restrictions, such as e. g. triangle-free graphs [41] and more generally graphs with no induced cycle of given length [35], cubic graphs [18] and more generally k -regular graphs [16], planar graphs [17]. It can be solved in polynomial time for various graph classes, such as e. g.:

- :: P_4 -free graphs [11] and more generally perfect graphs [21],
- :: claw-free graphs [13, 32, 36–39, 43],
- :: fork-free graphs [3, 27],
- :: apple-free graphs [7, 8],
- :: $2K_2$ -free graphs [14] and more generally lK_2 -free graphs for any constant l (by combining an algorithm generating all maximal independent sets of a graph [44] and a polynomial upper bound on the number of maximal independent sets in lK_2 -free graphs [2, 15, 42]),
- :: lP_3 -free graphs for any constant l [26] and more generally $lK_{1,3}$ -free graphs (i.e., l claw-free graphs) for any constant l [9],
- :: P_5 -free graphs [25],
- :: P_6 -free graphs [22].

Let us mention the (deep) recent results showing that WIS can be solved in sub-exponential time for P_t -free graphs for any fixed natural t [5, 10] (see also [24] for P_6 -free graphs) and in quasi-polynomial time for P_t -free graphs for any fixed natural t [20, 40].

As one can easily check, for any graph G one has

$$\alpha_w(G) = \max\{w(v) + \alpha_w(G[A(v)]) : v \in V\}$$

Then WIS for any graph G can be reduced to the same problem for the anti-neighborhoods of all vertices of G . Then one has:

Proposition 1 *For any graph F , if WIS can be solved for F -free graphs in polynomial time, then WIS can be solved for $lK_1 + F$ -free graphs in polynomial time for any constant l . □*

Let us report the following result due to Alekseev [1, 4]. Let us say that a graph is of type T if it is the graph $S_{i,j,k}$ depicted in Fig. 1 for some indices i, j, k (e.g., $S_{0,1,2}$ is a P_4 , $S_{1,1,1}$ is a claw).

Theorem 1 [1] *Let \mathcal{X} be a class of graphs defined by a finite set M of forbidden induced subgraphs. If M contains no graph every connected component of which is of type T , then WIS is NP-hard in the class \mathcal{X} . □*

Alekseev's result implies that WIS is NP-hard for $K_{1,4}$ -free graphs—the fact that WIS is NP-hard for $K_{1,4}$ -free graphs is already mentioned in [32].

Alekseev's result implies that (unless $P = NP$) for any graph F , if WIS is polynomial for F -free graphs, then each connected component of F is of type T . By Proposition 1, for any graph F , if WIS can be solved for F -free graphs in polynomial time, then WIS can be solved for $lK_1 + F$ -free graphs for any constant l in polynomial time. It follows that, since WIS can be solved in polynomial time for fork-free graphs [4, 27], for P_6 -free graphs [22, 24], and for l claw-free graphs for any constant l [9], the minimal graphs F of type T for which the complexity of WIS for F -free graphs is open are: P_7 , $S_{1,1,3}$, $S_{1,2,2}$, $K_2 + P_4$. Then the minimal graph classes, defined by forbidding one induced subgraph, for which the complexity of WIS is open are: P_7 -free graphs, $S_{1,1,3}$ -free graphs, $S_{1,2,2}$ -free graphs, $K_2 + P_4$ -free graphs.

The main result of this manuscript is the following:

Theorem 2 *For any constant m and for any graph family \mathcal{Y} containing an induced subgraph of $Y_{m,m}$, if WIS can be solved for \mathcal{Y} -free graphs in polynomial time, then WIS can be solved for $lK_2 + \mathcal{Y}$ -free graphs in polynomial time for any constant l . \square*

The proof of the main result is based on Farber's approach to prove that every $2K_2$ -free graph has $O(n^2)$ maximal independent sets [14], which directly leads to a polynomial time algorithm to solve WIS for $2K_2$ -free graphs through a dynamic programming approach, and on some extensions of Farber's approach [28, 29].

In details Sect. 2 reports Farber's approach, together with related results, and preliminaries; Sect. 3 reports the preliminary step from [29]; Sect. 4 provides the main step for the case of $K_2 + Y_{m,m}$ -free graphs; Sect. 5 provides the further step for the case of $lK_2 + Y_{m,m}$ -free graphs for any constant l by iterating the main step.

Then by Theorem 2 one obtains two corollaries for the case in which \mathcal{Y} is defined by a single graph say Y , that is $\mathcal{Y} = \{Y\}$, where Y is an induced subgraph of $Y_{m,m}$. They provide an extension of some polynomial results [and of some possibly forthcoming polynomial results] and which may be useful in view of a possible classification, according to Alekseev's result [1, 4], of the complexity of WIS for graph classes defined by forbidding one induced subgraph.

The first corollary concerns those Y -free graphs for which the complexity of (W)IS is known to be polynomial. According to the aforementioned known results, such Y -free graphs are fork-free graphs [i.e. $Y_{0,2}$ -free graphs] [3, 27], which include P_4 -free graphs [11] and claw-free graphs [13, 32, 36–39, 43], and P_5 -free graphs [i.e. $Y_{1,1}$ -free graphs] [25].

Corollary 1 *WIS can be solved for lK_2 +fork-free graphs and for $lK_2 + P_5$ -free graphs, for any constant l , in polynomial time. \square*

In particular Corollary 1: (i) addresses the open case concerning WIS for $K_2 + P_4$ -free graphs; (ii) extends the known polynomial results concerning WIS for fork-free graphs and for P_5 -free graphs; (iii) implies that the new minimal graph classes, defined by forbidding one induced subgraph, for which the complexity of (W)IS is open are: P_7 -free graphs, $S_{1,1,3}$ -free graphs, $S_{1,2,2}$ -free graphs, $K_2 + P_6$ -free graphs, $P_3 + P_4$ -free graphs.

The second corollary concerns those Y -free graphs for which the complexity of WIS is known to be open. According to the aforementioned known results and to Alekseev’s result [1, 4], such Y -free graphs are $S_{1,1,3}$ -free graphs [i.e. $Y_{1,2}$ -free graphs].

Corollary 2 *If WIS can be solved for $S_{1,1,3}$ -free graphs in polynomial time, then WIS can be solved for $lK_2 + S_{1,1,3}$ -free graphs for any constant l in polynomial time. □*

Remark 1 Concerning Corollary 1, let us recall that MIS can be solved for fork-free graphs in polynomial time with a not specified time bound [3]. By [27] WIS can be solved for fork-free in nT time where T is the time bound to solve WIS for claw-free graphs. Different references on polynomial time algorithm to solve WIS for claw-free graphs [32, 36, 43] were known, however as recently shown WIS can be solved for claw-free graphs in $O(n^3)$ time by [13], and even in $\mathcal{O}(n^2 \log n)$ time by [37] which is the best known time bound. Then by [27] WIS can be solved for fork-free graphs in $O(n^3 \log n)$ time. On the other hand, WIS can be solved for P_5 -free graphs G in $O(|E(G)||V(G)|^2)$ time [25]. □

Remark 2 Concerning Corollary 2, let us recall that MIS can be solved for $(S_{1,1,3}, K_{p,p})$ -free graphs (for any constant p) in polynomial time [12]; then, by Theorem 2, MIS can be solved for $lK_2 + (S_{1,1,3}, K_{p,p})$ -free graphs (for any constants l and p) in polynomial time. □

Remark 3 Let us observe that (a) the results of this manuscript are not implied by Corollary 6 of [29], since Corollary 6 of [29] concerns subclasses of $Y_{m,m}$ -free graphs, and (b) the results of this manuscript do not imply and should not be implied by the results of [34], since those results concern the class of m -plausible graphs for a constant m ; in particular, m -plausible graphs (e.g., chordal graphs) are not necessarily $Y_{m,m}$ -free graphs, and it seems to be difficult to directly show that $Y_{m,m}$ -free graphs are m -plausible graphs as remarked in [34]. □

Let us conclude this section by formalizing two possible open problems.

Open Problem 1 The complexity of WIS (of MIS) for the new minimal graph classes, defined by forbidding one induced subgraph, for which it is open: P_7 -free graphs, $S_{1,1,3}$ -free graphs, $S_{1,2,2}$ -free graphs, $K_2 + P_6$ -free graphs, $P_3 + P_4$ -free graphs.

Open Problem 2 New sufficient conditions to extend Proposition 1 to the case of lK_2 -free graphs for any constant l , or more generally, new extensions of lK_2 -free graphs for any constant l for which WIS (for which MIS) can be solved in polynomial time.

2 Independent Sets in $2K_2$ -free Graphs

In this section, let us report from [29] an algorithm, namely Algorithm Alpha, which formalizes the aforementioned Farber’s approach [14] and which is the basis of what follows.

For a graph G , with vertex set $\{v_1, v_2, \dots, v_n\}$, let us write $G_i := G[\{v_1, v_2, \dots, v_i\}]$. At each iteration i , $1 \leq i \leq n$, Algorithm Alpha provides a family \mathcal{S}_i of subsets of $\{v_1, v_2, \dots, v_i\}$ such that each maximal independent set of G_i is contained in some member of \mathcal{S}_i .

Algorithm Alpha

Input: A $2K_2$ -free graph G with vertex set $\{v_1, v_2, \dots, v_n\}$.

Output: A family \mathcal{S} of subsets of $V(G)$.

$\mathcal{S} := \{\emptyset\}$

For $i = 1, \dots, n$, do

begin

Step 1 [Extension of some members of \mathcal{S}]

For each $H \in \mathcal{S}$,

If $H \cup \{v_i\}$ is an independent set,

then $H := H \cup \{v_i\}$.

Step 2 [Addition of some members to \mathcal{S}]

For each K_2 of G_i containing v_i , i.e., for each edge uv_i of G_i ,

$H := \{v_i\} \cup A_{G_i}(u, v_i)$,

$\mathcal{S} := \mathcal{S} \cup \{H\}$.

end

Then let us consider the following algorithm to solve WIS for $2K_2$ -free graphs.

Algorithm WIS

Input: A $2K_2$ -free graph G .

Output: A maximum weight independent set of G .

PHASE 1. Execute Algorithm Alpha for G . Let \mathcal{S} be the family of subsets of G produced in this way.

PHASE 2. For each $H \in \mathcal{S}$, compute a maximum weight independent set of $G[H]$ (each $H \in \mathcal{S}$ is an independent set since G is $2K_2$ -free). Then choose a best solution, i.e., one of maximum weight.

It is clear from the description of Algorithm Alpha that, since G is $2K_2$ -free, every member of the family \mathcal{S} produced by the algorithm is an independent set of G . Moreover, according to Farber's argumentation, \mathcal{S} contains all maximal independent sets of G , which can be proven by induction on i . Then one obtains the following result:

Theorem 3 [14] *WIS can be solved for $2K_2$ -free graphs in $O(n^4)$ time via Algorithm WIS.* \square

Then let us report an extension of Farber's approach for lK_2 -free graphs for any constant l ; that is split (according to the literature) into a structure result shown by

different authors [2, 15, 42] and a computational result shown by combining such a structure result and an algorithm generating all maximal independent sets of a graph [44].

Theorem 4 [2, 15, 42] *For any constant l and for any lK_2 -free graph G , the family of all maximal independent sets of G contains $O(n^{2l-2})$ members.* □

Theorem 5 [2, 15, 42, 44] *For any constant l and for any lK_2 -free graph G one has: (i) the family of all maximal independent sets of G can be computed in polynomial time, (ii) WIS can be solved for G in polynomial time.* □

Farber’s approach seems to admit further possibilities of extension, e.g. [9, 15, 29–31, 33, 34], one of which is reported in Sect. 3.

Finally let us introduce a definition which will be useful later.

Notation: For convenience let us denote the star $K_{1,t}$, for any constant t , by the pair (c, L) , where c is the center of the star and L is the set of leaves of the star with $|L| = t$.

Definition 1 Let G be a graph and let p be a natural. A p -star set of G is a set $H \subset V(G)$ such that

- (i) G contains p induced stars, namely $(c_1, L_1), \dots, (c_p, L_p)$, called the *stars of H* , which have mutually a co-join,
- (ii) $H = L_1 \cup \dots \cup L_p \cup H_{anti}$, where H_{anti} is a subset of $A_G(c_1, \dots, c_p, L_1, \dots, L_p)$.

Furthermore let us say that, for any natural m , a p -star set of G , say $H = L_1 \cup \dots \cup L_p \cup H_{anti}$, is m -full if $|L_i| \geq m$ for every $i \in \{1, \dots, p\}$.

The following observation can be proved by the above definition.

Observation 1 Let G be a graph and let H be a p -star set of G for a natural p . Then all non-trivial components of $G[H]$ are contained in the anti-neighborhood of an (easily detectable) induced pK_2 of G . □

3 Independent Sets in $Y_{m,m}$ -free Graphs

In this section, let us present an algorithm, namely Algorithm Beta(0, m) where m is any fixed natural, which is just a slightly modified version of Algorithm $\mathcal{B}(m)$ of [29], which is an extension of Algorithm Alpha. That allows a very similar description between this algorithm and its extensions presented in the next sections, namely Algorithm Beta(l, m), for $l \geq 1$; in this sense the description of Algorithm Beta(0, m) may look redundant with respect to that of Algorithm $\mathcal{B}(m)$ of [29] but it already prepares the description of its extensions presented in the next sections.

The subsequent Algorithm Beta(0, m) computes a family \mathcal{S} of subsets of $V(G)$, for any input $Y_{m,m}$ -free graph G , which fulfills properties analogous to Lemma 3, Lemma 4, and Theorem 5 of [29].

Let us introduce some preliminaries for Algorithm Beta(0, m).

For a graph G , with vertex set $\{v_1, v_2, \dots, v_n\}$, let us write $G_i := G[\{v_1, v_2, \dots, v_i\}]$. At each iteration i , $1 \leq i \leq n$, Algorithm Beta(0, m) provides a family \mathcal{S}_i of subsets of $\{v_1, v_2, \dots, v_i\}$ such that each maximal independent set of G_i is contained in some member of \mathcal{S}_i .

The members of \mathcal{S} are of two kinds: 1-*marked* and *unmarked*.

Each 1-marked member is created and marked [or derives from a 1-marked member which has been created and marked] in Step 2 of some loop. Then it is possibly extended in Step 1 of the successive loops. **Each 1-marked member H is a 1-star set of G** whose star is of a center vertex $u_1(H) \notin H$ called the *marker* of H . Each 1-marked member H keeps its marker $u_1(H)$ along its possible extensions.

Each unmarked member (in particular the family \mathcal{S} produced by Algorithm Beta (0, m) contains exactly one unmarked member) is created in the initialization step as the empty set. Then it is possibly extended in Step 1 of the successive loops. **Each unmarked member is an independent set of G .**

For any subset H of $V(G)$, let us denote as:

H_0 the vertex-set of trivial components of $G[H]$ (i.e., the isolated vertices of H);

H^* the vertex-set of nontrivial components of $G[H]$;

$\mathcal{C}(G[H])$ the family of components of $G[H]$.

For any subset H of $V(G)$ and for any vertex $v \in V(G) \setminus H$, let us say that with respect to H :

- (1) v is of *Type 1* if v does not contact H ;
- (2) v is of *Type 2* if v dominates some member of $\mathcal{C}(G[H])$; *Note:* If v is not of Type 1 or Type 2, then v does not contact H_0 , v contacts H^* , v dominates no member of $\mathcal{C}(G[H])$. Type 3 and Type 4 are meaningful only if H is a 1-marked member of \mathcal{S} .
- (3) v is of *Type 3* if v is not of Type 1 or Type 2, v is nonadjacent to $u_1(H)$;
- (4) v is of *Type 4* if v is not of Type 1 or Type 2, v is adjacent to $u_1(H)$.

Algorithm Beta(0, m)

Input: A $Y_{m,m}$ -free graph G with vertex set $\{v_1, v_2, \dots, v_n\}$.

Output: A family \mathcal{S} of subsets of $V(G)$.

$\mathcal{S}_0 := \{\emptyset\}$ (and say that \emptyset is *unmarked*)

For $i = 1, \dots, n$, do

begin

$\mathcal{S}_i := \mathcal{S}_{i-1}$

Step 1 [Extension of some members of \mathcal{S}_i]

For each $H \in \mathcal{S}_{i-1}$, do

(1.1) If H is unmarked and if v_i is of Type 1,

then $H := H \cup \{v_i\}$ (and say that H remains *unmarked*)

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$

(1.2) If H is 1-marked and if v_i is of Type 1 or of Type 3,

then $H := H \cup \{v_i\}$ (and say that H remains *1-marked*)

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$

Step 2 [Addition of 1-marked members to \mathcal{S}_i]

For each induced star (c, L) of G_i such that $|L| \leq m$,

$H := L \cup A_{G_i}(c, L)$,

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$ (and say that H is 1-marked and that the marker of H is $u_1(H) := c$)

end

$\mathcal{S} := \mathcal{S}_i$

end.

The following lemmas can be proved by arguments similar to that of Lemma 3, Lemma 4, and Theorem 5 of [29]; for the sake of completeness let us describe such arguments.

Lemma 1 Referring to Algorithm Beta(0,m):

- (i) every 1-marked member H of \mathcal{S}_i , for $i = 1, \dots, n$, is a 1-star set whose star is of center $u_1(H)$ and of leaves a subset of H_0 ;
- (ii) every unmarked member of \mathcal{S}_i , for $i = 1, \dots, n$, is an independent set (in particular \mathcal{S}_i contains exactly one unmarked member);
- (iii) the family \mathcal{S} produced by Algorithm Beta(0,m) can be partitioned into $\{\mathcal{S}_{one}, \mathcal{S}_{two}, \mathcal{S}_{three}\}$, where: $\mathcal{S}_{one} = \{I \in \mathcal{S} : I \text{ is an independent set}\}$, $\mathcal{S}_{two} = \{I \in \mathcal{S} : I \text{ is a 1-star } m\text{-full set, } I \text{ is not an independent set}\}$, $\mathcal{S}_{three} = \{I \in \mathcal{S} : I \text{ is a 1-star set, } I \text{ is not an independent set, } I \text{ is not a 1-star } m\text{-full set}\}$.

Proof Proof of (i). Each 1-marked member is created in Step 2 of some loop and then it is possibly extended in Step (1.2) of the successive loops. Then, by definition of Step 2 and by definition of $u_1(H)$, each 1-marked member H satisfies Statement (i) at the moment of its creation. Furthermore each 1-marked member satisfies Statement (i) after its possible extensions by definition of a vertex of Type 1 and Type 3 (let us observe that each 1-marked member H keeps its marker $u_1(H)$ along its possible extensions).

Proof of (ii). If an unmarked member has been created as an independent set, then it will remain an independent set after its possible extensions in Step (1.1) by definition of a vertex of Type 1. Then, to complete the proof, note that exactly one unmarked member is created, i.e., the empty set in the initialization step, and the empty set is trivially an independent set.

Prof of (iii). It follows by Statements (i)–(ii) and by definition of Algorithm Beta (0,m). □

Lemma 2 Let \mathcal{S} be the family of subsets of $V(G)$ produced by Algorithm Beta(0,m). Then for any maximal independent set I of G there is a member $H \in \mathcal{S}$ such that $I \subseteq H$.

Proof As above let \mathcal{S}_i denote the content of the family \mathcal{S} after i loops of the algorithm, for $i = 1, \dots, n$. Then to prove the lemma, let us show that for any

maximal independent set I of G_i , there is a member $H \in \mathcal{S}_i$ such that $I \subseteq H$. The proof is by induction on i , for $i = 1, \dots, n$.

For $i = 1$, $\mathcal{S}_1 = \{\{v_1\}\}$, and $\{v_1\}$ is obviously the only maximal independent set in the graph G_1 . Then let us assume that the assertion holds for $i - 1$ and prove that it holds for i .

Let I be a maximal independent set of G_i .

Assume that $v_i \notin I$. Then by the inductive hypothesis I is contained in some member of \mathcal{S}_{i-1} and thus of \mathcal{S}_i , since each member of \mathcal{S}_{i-1} is contained (properly or not) in some member of \mathcal{S}_i .

Assume that $v_i \in I$. Then by the inductive hypothesis let H be a member of \mathcal{S}_{i-1} containing $I \setminus \{v_i\}$.

Case 1. H is unmarked.

By Lemma 1 (ii) H is an independent set, i.e., $H = H_0$, and thus v_i can be just of Type 1 or 2 with respect to H . If v_i is of Type 1 with respect to H , then $I \subseteq H \cup \{v_i\} \in \mathcal{S}_i$ (Step 1). If v_i is of Type 2 with respect to H , then there exists a vertex $u \in N_{G_i}(v_i)$ that has no neighbors in $I \setminus \{v_i\}$, that is, I is contained in the set $\{v_i\} \cup A_{G_i}(u, v_i) \in \mathcal{S}_i$ (Step 2).

Case 2. H is 1-marked.

If v_i is of Type 1 or of Type 3 with respect to H , then $I \subseteq H \cup \{v_i\} \in \mathcal{S}_i$ (Step 1). If v_i is of Type 2 with respect to H , then there exists a vertex $u \in N_{G_i}(v_i)$ that has no neighbors in $I \setminus \{v_i\}$, that is, I is contained in the set $\{v_i\} \cup A_{G_i}(u, v_i) \in \mathcal{S}_i$ (Step 2).

Finally let us consider the case in which v_i is of Type 4 with respect to H .

Since v_i is of Type 4, v_i is adjacent to $u_1(H)$ and to a vertex say h of H^* . Let us write $U = N(u_1(H)) \cap (I \setminus \{v_i\})$ and $T = N(h) \cap (I \setminus \{v_i\})$. Note that, by Lemma 1 (i), $\{u_1(H)\} \cup U$ has a co-join to $\{h\} \cup T$. Furthermore, by construction, $U \neq \emptyset$.

Assume that $T = \emptyset$. Then I is contained in the set $\{v_i\} \cup A_{G_i}(h, v_i) \in \mathcal{S}_i$ (Step 2).

Assume that $T \neq \emptyset$. Therefore, either $|U| < m$, or $|T| < m$, else vertices of $\{u_1(H)\} \cup U \cup \{v_i\} \cup \{h\} \cup T$ induce a $Y_{m,m}$, which is not possible since G is $Y_{m,m}$ -free. Then I is contained in the set $L \cup A_{G_i}(c, L)$ (Step 2), where either $c = u_1(H)$ and $L = \{v_i\} \cup U$, or $c = h$ and $L = \{v_i\} \cup T$. □

Lemma 3 Algorithm Beta(0,m) runs in time $O(n^{m+5})$ and computes a family \mathcal{S} of size $O(n^{m+2})$.

Proof Concerning 1-marked members of \mathcal{S} . The total number of new 1-marked members added to \mathcal{S} by each Step 2 of each loop of Algorithm Beta(0, m) is at most n^{m+2} (that is at most n^{m+1} for each loop). In particular, since each 1-marked member can be created/extended in $O(n^2)$ time (in particular in order to determine the connected components) and since the loops are n , all 1-marked members can be computed in $O(n^{m+5})$ time.

Concerning unmarked members of \mathcal{S} . Exactly one unmarked member is contained in \mathcal{S} . Such an unmarked member is created as the empty set in the initialization step and then it is possibly extended in Step 1 of the successive loops of Algorithm Beta (0, m) in $O(n)$ time; therefore, since the loops are n , it can be computed in $O(n^2)$ time.

Therefore $\max\{O(n^{m+5}), O(n^2)\}$ ($= O(n^{m+5})$) is the time bound of Algorithm Beta(0, m). \square

Then, let us consider the following algorithm to solve WIS for \mathcal{Y} -free graphs, where \mathcal{Y} is any graph family containing an induced subgraph of $Y_{m,m}$.

Algorithm WIS(0, m)

Input: A \mathcal{Y} -free graph G .

Output: A maximum weight independent set of G .

PHASE 1. Execute Algorithm Beta(0, m) for G . Let \mathcal{S} be the family of subsets of G produced in this way.

PHASE 2. For each $H \in \mathcal{S}$, compute a maximum weight independent set of $G[H]$. Then choose a best solution, i.e., one of maximum weight.

Theorem 6 *Algorithm WIS(0, m) is correct and can be executed in polynomial time whenever PHASE 2 can be executed in polynomial time.*

Proof *Correctness.* By Lemma 2, Algorithm WIS(0, m) is correct.

Complexity. By Lemma 3, PHASE 1 can be executed in polynomial time. Then Algorithm WIS(0, m) can be executed in polynomial time whenever PHASE 2 can be executed in polynomial time. \square

4 Independent Sets in $K_2 + Y_{m,m}$ -free Graphs

In this section, let us introduce an extension of Algorithm Beta(0, m) of the previous section.

The subsequent Algorithm Beta(1, m) computes a family \mathcal{S} of subsets of $V(G)$, for any input $K_2 + Y_{m,m}$ -free graph G , which fulfills properties analogous to Lemmas 1, 2, 3.

The preliminaries for Algorithm Beta(1, m) are exactly those for Algorithm Beta(0, m) except for the following points.

The members of \mathcal{S} are of three kinds: 1-marked, 2-marked, unmarked.

Each 1-marked member is created and marked [or derives from a 1-marked member which has been created and marked] in Step 3 of some loop. Then it is possibly extended in Step 1 of the successive loops. **Each 1-marked member H is a 1-star set of G** whose star is of a center vertex $u_1(H) \notin H$ called the *marker* of H . Each 1-marked member H keeps its marker $u_1(H)$ along its possible extensions.

Each 2-marked member is created with respect to some 1-marked member in Step 2 of some loop. Then it is possibly extended in Step 1 of the successive loops. **Each 2-marked member H is a 2-star m -full set of G** whose stars are respectively of center two vertices $u_1(H), u_2(H) \notin H$ called the *markers* of H . Each 2-marked member H keeps its markers $u_1(H), u_2(H)$ along its possible extensions.

Each unmarked member is created either in the initialization step, i.e., the empty set, or in Step 2 of some loop with respect to some p -marked member for some

$p \in \{1, 2\}$. Then it is possibly extended in Step 1 of the successive loops. **Each unmarked member is an independent set of G .**

For any subset H of $V(G)$ and for any vertex $v \in V(G) \setminus H$, let us say that with respect to H :

- (1) v is of *Type 1* if v does not contact H ;
- (2) v is of *Type 2* if v dominates some member of $\mathcal{C}(G[H])$; *Note:* If v is not of Type 1 or Type 2, then v does not contact H_0 , v contacts H^* , v dominates no element of $\mathcal{C}(G[H])$. Type 3 and Type 4 are meaningful only if H is a 1-marked or a 2-marked member of \mathcal{S} . Type 5 is meaningful only if H is a 2-marked member of \mathcal{S} . In the definition of Type 5 the term 'at least' is added in order to save such group of definitions for the next section.
- (3) v is of *Type 3* if v is not of Type 1 or Type 2, v is adjacent to no marker of H ;
- (4) v is of *Type 4* if v is not of Type 1 or Type 2, v is adjacent to exactly one marker of H ;
- (5) v is of *Type 5* if v is not of Type 1 or Type 2, v is adjacent to at least two markers of H .

Algorithm Beta(1, m)

Input: A $K_2 + Y_{m,m}$ -free graph G with vertex set $\{v_1, v_2, \dots, v_n\}$.

Output: A family \mathcal{S} of subsets of $V(G)$.

$\mathcal{S}_0 := \{\emptyset\}$ (and say that \emptyset is *unmarked*)

For $i = 1, \dots, n$, do

begin

$\mathcal{S}_i := \mathcal{S}_{i-1}$

Step 1 [Extension of some members of \mathcal{S}_i]

For each $H \in \mathcal{S}_{i-1}$, do

(1.1) If H is unmarked and if v_i is of Type 1,

then $H := H \cup \{v_i\}$ (and say that H remains *unmarked*)

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$

(1.2) If H is p -marked for some $p \in \{1, 2\}$,

and if v_i is either of Type 1 being adjacent to at most one marker of H or of Type 3,

then $H := H \cup \{v_i\}$ (and say that H remains *p-marked*)

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$

Step 2 [Addition of unmarked members and of 2-marked members to \mathcal{S}_i]

For each $H \in \mathcal{S}_{i-1}$, do

(2.1) If H is 1-marked,

if $u_1(H)$ has at least m neighbors in H_0 ,

and if v_i is of Type 4,

then

:: execute Algorithm Beta(0, m) for $G[H^* \setminus N(v_i)]$,

:: let \mathcal{S}' be the family of subsets of $V(G[H^* \setminus N(v_i)])$ produced in this way,

:: let $\mathcal{S}'_{one} = \{M' \in \mathcal{S}' : M' \text{ is an independent set}\}$,

:: let $\mathcal{S}'_{two} = \{M' \in \mathcal{S}' : M' \text{ is a 1-star } m\text{-full set, } M' \text{ is not an independent set}\}$

:: $\mathcal{Q}(i, H) := \{\{v_i\} \cup H_0 \cup M' : M' \in \mathcal{S}'_{one}\}$,

:: $\mathcal{S}_i := \mathcal{S}_i \cup \mathcal{Q}(i, H)$ (and say that each member of $\mathcal{Q}(i, H)$ is *unmarked*)

:: $\mathcal{T}(i, H) := \{\{v_i\} \cup H_0 \cup M' : M' \in \mathcal{S}'_{two} \text{ whose star is of center say } c'_1\}$,

:: $\mathcal{S}_i := \mathcal{S}_i \cup \mathcal{T}(i, H)$

(and say that each member $H' = \{v_i\} \cup H_0 \cup M' \in \mathcal{T}(i, H)$ is *2-marked*

and that the markers of H' are $u_1(H') := u_1(H)$, $u_2(H') := c'_1$).

(2.2) If H is 2-marked,

and if v_i is either of Type 1 being adjacent to at least two markers of H or of Type 5, then

:: compute the family \mathcal{R} of all maximal independent sets of $G[H^* \setminus N(v_i)]$

:: $\mathcal{R}(i, H) := \{\{v_i\} \cup H_0 \cup R : R \in \mathcal{R}\}$

:: $\mathcal{S}_i := \mathcal{S}_i \cup \mathcal{R}(i, H)$ (and say that each member of $\mathcal{R}(i, H)$ is *unmarked*)

Step 3 [Addition of 1-marked members to \mathcal{S}_i]

For each induced star (c, L) of G_i such that $|L| \leq m$,

$H := L \cup A_{G_i}(c, L)$,

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$ (and say that H is *1-marked* and that the marker of H is $u_1(H) := c$)

end

$\mathcal{S} := \mathcal{S}_i$

end.

Let us prove the following lemmas.

Note: Statement (i') of the subsequent Lemma 4 ensures that Step (2.1) of Algorithm Beta(1, m) is well defined, in fact $G[H^* \setminus N(v_i)]$ needs to be $Y_{m,m}$ -free in order to execute Algorithm Beta(0, m) for $G[H^* \setminus N(v_i)]$.

Lemma 4 Referring to Algorithm Beta(1, m):

- (i) every 1-marked member H of \mathcal{S}_i , for $i = 1, \dots, n$, is a 1-star set whose star is of center $u_1(H)$ and of leaves a subset of H_0 ;
- (ii) every 2-marked member H of \mathcal{S}_i , for $i = 1, \dots, n$, is a 2-star m -full set whose stars respectively are of center $u_1(H), u_2(H)$ and of leaves a subset of H_0 ;
- (iii) referring to Step (2.1) of Loop i of Algorithm Beta(1, m), for $i = 1, \dots, n$, $G[H^* \setminus N(v_i)]$ is $Y_{m,m}$ -free;
- (iv) referring to Step (2.2) of Loop i of Algorithm Beta(1, m), for $i = 1, \dots, n$, $G[H^* \setminus N(v_i)]$ is K_2 -free, i.e., $H^* \setminus N(v_i)$ is an independent set;
- (v) every unmarked member of \mathcal{S}_i , for $i = 1, \dots, n$, is an independent set;
- (vi) the family \mathcal{S} produced by Algorithm Beta(1, m) can be partitioned into $\{\mathcal{S}_{one}, \mathcal{S}_{two}, \mathcal{S}_{three}\}$, where: $\mathcal{S}_{one} = \{I \in \mathcal{S} : I \text{ is an independent set}\}$, $\mathcal{S}_{two} = \{I \in \mathcal{S} : I \text{ is a } p\text{-star } m\text{-full set for some } p \in \{1, 2\}, I \text{ is not an independent set}\}$, $\mathcal{S}_{three} = \{I \in \mathcal{S} : I \text{ is a } p\text{-star set for some } p \in \{1, 2\}, I \text{ is not an independent set, } I \text{ is not a } p\text{-star } m\text{-full set for } p \in \{1, 2\}\}$.

Proof Proof of (i). The proof is similar to that of Lemma 1 (i), with Step 3 instead of Step 2.

Proof of (ii). Each 2-marked member is created in Step (2.1) of some loop and then it is possibly extended in Step (1.2) of the successive loops. Then, by definition of Step (2.1) and by definition of $u_1(H)$ and $u_2(H)$, each 2-marked member satisfies Statement (ii) at the moment of its creation. Furthermore each 2-marked member satisfies Statement (ii) after its possible extensions by definition of a vertex of Type 1 being adjacent to at most one marker of H and of Type 3 (let us observe that each 2-marked member H keeps its markers $u_1(H), u_2(H)$ along its possible extensions).

Proof of (i'). At Step (2.1) of Loop i of Algorithm Beta(1, m), for $i = 1, \dots, n$, by construction $H^* \setminus N(v_i)$ is contained in the anti-neighborhood of $\{v_i\} \cup u_1(H) \cup (N(u_1) \cap H_0)$, which is a star of center $u_1(H)$ (thus containing a K_2) by Statement (i) and since v_i is of Type 4. Then Statement (i') follows since G is $K_2 + Y_{m,m}$ -free.

Proof of (ii'). At Step (2.2) of Loop i of Algorithm Beta(1, m), for $i = 1, \dots, n$, H is 2-marked and v_i is either of Type 1 being adjacent to at least two markers of H or of Type 5. Therefore, by Statement (ii), v_i and the two stars of H induce a subgraph of G_i that contains an induced $Y_{m,m}$. Then Statement (ii') follows since G is $K_2 + Y_{m,m}$ -free.

Proof of (iii). If an unmarked member has been created as an independent set, then it will remain an independent set after its possible extensions in Step (1.1) by definition of a vertex of Type 1. Then, to complete the proof, note that unmarked members are created either in the initialization step, i.e., the empty set, or in Step 2 of each loop, which is well defined by Statement (i'). Concerning the empty set, it is trivially an independent set. Concerning Step 2 of each loop: in Step (2.1) unmarked members (i.e., the members of $\mathcal{Q}(i, H)$) are independent sets by definition of H_0 , of H^* , and of S'_{one} ; in Step (2.2) unmarked members (i.e., the members of $\mathcal{R}(i, H)$) are independent sets by definition.

Proof of (iv). It follows by Statements (i)–(iii) and by definition of Algorithm Beta (1, m). \square

Lemma 5 *Let \mathcal{S} be the family of subsets of $V(G)$ produced by Algorithm Beta(1, m). Then for any maximal independent set I of G , there is a member $H \in \mathcal{S}$ such that $I \subseteq H$.*

Proof The first part of this proof is similar to the first part of the proof of Lemma 2.

Case 1. H is unmarked.

This case is similar to Case 1 of Lemma 2, with Lemma 4 instead of Lemma 1, and with Step 3 instead of Step 2.

Case 2. H is p -marked for some $p \in \{1, 2\}$.

Case 2.1. $p = 1$.

By Lemma 4 (i) H is a 1-star set of G whose star is of center $u_1(H)$ and of leaves a subset of H_0 . In particular v_i can be just of Type 1 or 2 or 3 or 4 with respect to H . If v_i is of Type 1 with respect to H being adjacent to at most one marker of H or of Type 3 with respect to H , then $I \subseteq H \cup \{v_i\} \in \mathcal{S}_i$ (Step 1). If v_i is of Type 2 with respect to H , then there exists a vertex $u \in N_{G_i}(v_i)$ that has no neighbors in $I \setminus \{v_i\}$, that is, I is

contained in the set $\{v_i\} \cup A_{G_i}(u, v_i) \in \mathcal{S}_i$ (Step 3).

Let us consider the case in which v_i is of Type 1 with respect to H being adjacent to at least two markers of H .

Then I is contained in some member of the family $\mathcal{R}(i, H)$ generated by Step (2.2) of Loop i of Algorithm Beta(1, m).

Let us consider the case in which v_i is of Type 4 with respect to H .

Since I is a maximal independent set of G_i , I is a maximal independent set of $G[H \cup \{v_i\}]$ as well, i.e., $I \setminus \{v_i\}$ is a maximal independent set of $G[H \setminus N(v_i)]$. In particular, since v is of Type 4, one has that v_i is adjacent to $u_1(H)$ and that $H_0 \subset I \setminus \{v_i\}$.

If H is not m -full, i.e., $|N(u_1(H)) \cap H_0| < m$, then I is contained in the set $\{v_i\} \cup (N(u_1(H)) \cap H_0) \cup A_{G_i}(u_1(H), \{v_i\} \cup (N(u_1(H)) \cap H_0)) \in \mathcal{S}_i$ (Step 3).

If H is m -full, i.e., $|N(u_1(H)) \cap H_0| \geq m$, then let us consider the following argument.

Let us write $I = \{v_i\} \cup H_0 \cup I'$, where $I' = (I \setminus \{v_i\}) \cap (H^* \setminus N(v_i))$. Since $I \setminus \{v_i\}$ is a maximal independent set of $G[H \setminus N(v_i)]$, I' is a maximal independent set of $G[H^* \setminus N(v_i)]$. Now, according to Step (2.1) of Algorithm Beta(1, m) and according to Lemma 1 (iii), let $\mathcal{S}' = \mathcal{S}'_{one} \cup \mathcal{S}'_{two} \cup \mathcal{S}'_{three}$ be the family of subsets of $V(G[H^* \setminus N(v_i)])$ produced by Algorithm Beta(0, m) executed for $G[H^* \setminus N(v_i)]$. Therefore, by Lemma 2 there exists an $M' \in \mathcal{S}'$ such that $I' \subseteq M'$, and by Lemma 1 (iii) one of the following three subcases occurs.

Assume that $M' \in \mathcal{S}'_{one}$. Then $\{v_i\} \cup H_0 \cup M'$ belongs to the family $\mathcal{Q}(i, H)$ generated by Step (2.1) of Loop i of Algorithm Beta(1, m), i.e., I is contained in $\{v_i\} \cup H_0 \cup M' \in \mathcal{S}_i$.

Assume that $M' \in \mathcal{S}'_{two}$. Then $\{v_i\} \cup H_0 \cup M'$ belongs to the family $\mathcal{T}(i, H)$ generated by Step (2.1) of Loop i of Algorithm Beta(1, m), i.e., I is contained in $\{v_i\} \cup H_0 \cup M' \in \mathcal{S}_i$.

Assume that $M' \in \mathcal{S}'_{three}$. Therefore, M' is a 1-star set of $G[H^* \setminus N(v_i)]$, whose star say (c', L') is such that $|L'| < m$ and is clearly a star of G_i as well. Then I is contained in the set $L' \cup A_{G_i}(c', L') \in \mathcal{S}_i$ (Step 3).

Case 2.2. $p = 2$.

By Lemma 4 (ii) H is a 2-star m -full set of G whose stars respectively are of centers $u_1(H), u_2(H)$ and of leaves a subset of H_0 . If v_i is of Type 1 with respect to H being adjacent to at most one marker of H or of Type 3 with respect to H , then $I \subseteq H \cup \{v_i\} \in \mathcal{S}_i$ (Step 1). If v_i is of Type 2 with respect to H , then there exists a vertex $u \in N_{G_i}(v_i)$ that has no neighbors in $I \setminus \{v_i\}$, that is, I is contained in the set $\{v_i\} \cup A_{G_i}(u, v_i) \in \mathcal{S}_i$ (Step 3).

Let us consider the case in which v_i is either of Type 1 with respect to H being adjacent to at least two markers of H or of Type 5 with respect to H .

Then I is contained in some member of the family $\mathcal{R}(i, H)$ generated by Step (2.2) of Loop i of Algorithm Beta(1, m).

Let us consider the case in which v_i is of Type 4 with respect to H .

Without loss of generality by symmetry let us assume that v_i is adjacent to $u_1(H)$ and is non-adjacent to $u_2(H)$.

Let U_j , for $j = 1, 2$, denote the set of neighbors of $u_j(H)$ in H_0 ; then by the above $|U_j| \geq m$.

Since v_i is of Type 4, v_i is adjacent to a vertex say h of H^* . Let us write $T = N(h) \cap (I \setminus \{v_i\})$. Note that, by Lemma 4 (ii), $\{u_1(H)\} \cup U_1$ has a co-join to $\{h\} \cup T$.

Assume that $T = \emptyset$. Then I is contained in the set $\{v_i\} \cup A_{G_i}(h, v_i) \in \mathcal{S}_i$ (Step 3).

Assume that $T \neq \emptyset$. Therefore $|T| < m$, else vertices of $\{u_1(H)\} \cup U_1 \cup \{v_i\} \cup \{h\} \cup T$ would induce a $Y_{m,m}$, that is, any edge of the remaining star of H of center $u_2(H)$ and such an induced $Y_{m,m}$ would induce a $K_2 + Y_{m,m}$, which is not possible since G is $K_2 + Y_{m,m}$ -free. Then I is contained in the set $\{v_i\} \cup T \cup A_{G_i}(h, \{v_i\} \cup T) \in \mathcal{S}_i$ (Step 3). \square

The next lemma deals with the families $\mathcal{Q}(i, H)$ (unmarked members), $\mathcal{T}(i, H)$ (2-marked members), and $\mathcal{R}(i, H)$ (unmarked members) computed in Step 2 of Loop i of Algorithm Beta(1, m).

Lemma 6 Referring to Algorithm Beta(1, m):

- (i) for $i = 1, \dots, n$, and for any 1-marked member H , the family $\mathcal{Q}(i, H) \cup \mathcal{T}(i, H)$ contains $O(n^{m+1})$ members and can be computed in $O(n^{m+4})$ time;
- (ii) for $i = 1, \dots, n$, and for any 2-marked member H , the family $\mathcal{R}(i, H)$ contains one member and can be computed in $O(n)$ time;
- (iii) Algorithm Beta(1, m) runs in time $O(n^{2m+9})$ time and computes a family \mathcal{S} of size $O(n^{2m+5})$.

Proof Proof of (i): It follows by definition of Step (2.1), i.e., the execution of Algorithm Beta(0, m), and by Lemma 3.

Proof of (ii): It follows by definition of Step (2.2); in fact, by Lemma 4 (ii'), $H^* \setminus N(v_i)$ is an independent set; therefore $H \setminus N(v_i)$ is an independent set by definition of H_0 ; then the family $\mathcal{R}(i, H)$ contains one member and can be computed in $O(n)$ time.

Proof of (iii):

Concerning 1-marked members of \mathcal{S} . The total number of new 1-marked members added to \mathcal{S} by each Step 3 of each loop of Algorithm Beta(1, m) is at most n^{m+2} (that is at most n^{m+1} for each loop); in particular, since each 1-marked member can be created/extended in $O(n^2)$ time (in particular in order to determine the connected components) and since the loops are n , all 1-marked members can be computed in $O(n^{m+5})$ time.

Concerning 2-marked members of \mathcal{S} . Each 2-marked member is created with respect to some 1-marked member in Step (2.1) and then it is possibly extended in Step 1 of the successive loops. By statement (i), for $i = 1, \dots, n$, and for any 1-marked member H , the family $\mathcal{T}(i, H)$ of 2-marked members contains $O(n^{m+1})$ members and can be computed in $O(n^{m+4})$ time. Therefore, since $i = 1, \dots, n$ and since the total number of 1-marked members is $O(n^{m+2})$, the total number of 2-marked members from families $\mathcal{T}(i, H)$ (for $i = 1, \dots, n$, and for any 1-marked member H) is $O(n^{2m+4})$, and all such members can be computed in $O(n^{2m+7})$ time; in

particular, since each 2-marked member can be extended in $O(n)$ time and since the loops are n , all 2-marked members can be computed in $O(n^{2m+9})$ time.

Concerning unmarked members of \mathcal{S} . Each unmarked member (except for the unmarked member which is created as the empty set in the initialization step and then is possibly extended in Step 1 of the successive loops) is created with respect to some 1-marked member in Step (2.1) or with respect to some 2-marked member in Step (2.2), and then it is possibly extended in Step 1 of the successive loops.

By statement (i), for $i = 1, \dots, n$, and for any 1-marked member H , the family $\mathcal{Q}(i, H)$ of unmarked members contains $O(n^{m+1})$ members and can be computed in $O(n^{m+4})$ time. Therefore, since $i = 1, \dots, n$ and since the total number of 1-marked members is $O(n^{m+2})$, the total number of unmarked numbers from families $\mathcal{Q}(i, H)$ (for $i = 1, \dots, n$, and for any 1-marked member H) is $O(n^{2m+4})$, and all such members can be computed in $O(n^{2m+7})$ time; in particular, since each unmarked member can be extended in $O(n)$ time and since the loops are n , all such unmarked members can be computed in $O(n^{2m+9})$ time.

By statement (ii), for $i = 1, \dots, n$, and for any 2-marked member H , the family $\mathcal{R}(i, H)$ of unmarked members contains 1 member and can be computed in $O(n)$ time. Therefore, since $i = 1, \dots, n$ and since the total number of 2-marked members is $O(n^{2m+4})$ by the above, the total numbers of unmarked members from families $\mathcal{R}(i, H)$ (for $i = 1, \dots, n$, and for any 2-marked member H) is $O(n^{2m+5})$, and all such members can be computed in $O(n^{2m+6})$ time; in particular, since each unmarked member can be extended in $O(n)$ time and since the loops are n , all such unmarked members can be computed in $O(n^{2m+8})$ time.

Summarizing: the total number of members of the family \mathcal{S} produced by Algorithm Beta(1, m) is $O(n^{2m+5})$, and all such members can be computed in $O(n^{2m+9})$ time, which is also the time bound of Algorithm Beta(1, m). \square

Then, let us consider the following algorithm to solve WIS for $K_2 + \mathcal{Y}$ -free graphs, where \mathcal{Y} is any graph family containing an induced subgraph of $Y_{m,m}$.

Algorithm WIS(1, m)

Input: A $K_2 + \mathcal{Y}$ -free graph G .

Output: A maximum weight independent set of G .

PHASE 1. Execute Algorithm Beta(1, m) for G . Let \mathcal{S} be the family of subsets of G produced in this way.

PHASE 2. For each $H \in \mathcal{S}$, compute a maximum weight independent set of $G[H]$. Then choose a best solution, i.e., one of maximum weight.

Theorem 7 *Algorithm WIS(1, m) is correct and can be executed in $O(n^{2m+\max\{9,5+t\}})$ time whenever WIS can be solved for \mathcal{Y} -free graphs in $O(n^t)$ time for some constant t .*

Proof *Correctness.* By Lemma 5, Algorithm WIS(1, m) is correct.

Complexity.

Concerning PHASE 1: By Lemma 6, PHASE 1 can be executed in $O(n^{2m+9})$ time.

Concerning PHASE 2: By Lemma 6, the family \mathcal{S} contains $O(n^{2m+5})$ members.

On the other hand, by Lemma 4 (iv) and by Observation 1, for each $H \in \mathcal{S}$ only two cases may occur: either H is an independent set (and then the problem is trivial), or each non-trivial component K of $G[H]$ is contained in the anti-neighborhood of one edge of G (easily detectable), i.e., K is \mathcal{Y} -free since G is $K_2 + \mathcal{Y}$ -free. Therefore PHASE 2 can be executed in $O(n^{2m+5+t})$ time whenever WIS can be solved for \mathcal{Y} -free graphs in $O(n^t)$ time for some constant t .

Then Algorithm WIS(1, m) can be executed in $O(n^{2m+\max\{9,5+t\}})$ time whenever WIS can be solved for \mathcal{Y} -free graphs in $O(n^t)$ time for some constant t . \square

Then one obtains the following result.

Theorem 8 *For any constant m and for any graph family \mathcal{Y} containing an induced subgraph of $Y_{m,m}$, if WIS can be solved for \mathcal{Y} -free graphs in polynomial time, then WIS can be solved for $K_2 + \mathcal{Y}$ -free graphs in polynomial time.* \square

5 Independent Sets in $IK_2 + Y_{m,m}$ -free Graphs

In this section, let us introduce a class of algorithms which generalize/iterate Algorithm Beta(1, m) of the previous section.

The subsequent recursive Algorithm Beta(l , m), for any $l \geq 1$, computes a family \mathcal{S} of subsets of $V(G)$, for any input $IK_2 + Y_{m,m}$ -free graph G , which fulfills properties analogous to that of Lemmas 4, 5, 6. In particular Algorithm Beta(l , m), for $l = 1$, is exactly Algorithm Beta(1, m) of the previous section.

The preliminaries for Algorithm Beta(l , m) are exactly those for Algorithm Beta(1, m) except for the following points.

The members of \mathcal{S} are of $l + 2$ kinds: p -marked, for $p \in \{1, \dots, l + 1\}$, unmarked.

Each 1-marked member is created and marked [or derives from a 1-marked member which has been created and marked] in Step 3 of some loop. Then it is possibly extended in Step 1 of the successive loops. **Each 1-marked member H is a 1-star set of G** whose star is of a center vertex $u_1(H) \notin H$ called the *marker* of H . Each 1-marked member H keeps its marker $u_1(H)$ along its possible extensions.

Each p -marked member for $p \in \{2, \dots, l + 1\}$ is created with respect to some p' -marked member, with $p' < p$, in Step 2 of some loop. Then it is possibly extended in Step 1 of the successive loops. **Each p -marked member H is a p -star m -full set of G** whose stars are respectively of center p vertices $u_1(H), \dots, u_p(H) \notin H$ called the *markers* of H . Each p -marked member H keeps its markers $u_1(H), \dots, u_p(H)$ along its possible extensions.

Each unmarked member is created either in the initialization step, i.e., the empty set, or in Step 2 of some loop with respect to some p -marked member for some $p \in \{1, \dots, p\}$. Then it is possibly extended in Step 1 of the successive loops. **Each unmarked member is an independent set of G .**

Algorithm Beta(l, m)

Input: A $IK_2 + Y_{m,m}$ -free graph G and a vertex-ordering (v_1, v_2, \dots, v_n) of G .

Output: A family \mathcal{S} of subsets of $V(G)$.

$\mathcal{S}_0 := \{\emptyset\}$ (and say that \emptyset is *unmarked*)

For $i = 1, \dots, n$, do

begin

$\mathcal{S}_i := \mathcal{S}_{i-1}$

Step 1 [Extension of some members of \mathcal{S}_i]

For each $H \in \mathcal{S}_{i-1}$, do

(1.1) If H is unmarked and if v_i is of Type 1,

then $H := H \cup \{v_i\}$ (and say that H remains *unmarked*)

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$

(1.2) If H is p -marked for some $p \in \{1, \dots, l + 1\}$,

and if v_i is either of Type 1 being adjacent to at most one marker of H or of Type 3,

then $H := H \cup \{v_i\}$ (and say that H remains p -marked)

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$

Step 2 [Addition of unmarked members and of p -marked members for $p \in \{2, \dots, l + 1\}$ to \mathcal{S}_i]

For each $H \in \mathcal{S}_{i-1}$, do

(2.1) If H is p' -marked, for some $p' \in \{1, \dots, l\}$,

if $u_j(H)$ for $j \in \{1, \dots, p'\}$ has at least m neighbors in H_0 ,

and if v_i is of Type 4,

then

:: execute Algorithm Beta($l - p', m$) for $G[H^* \setminus N(v_i)]$,

:: let \mathcal{S}' be the family of subsets of $V(G[H^* \setminus N(v_i)])$ produced in this way,

:: let $\mathcal{S}'_{one} = \{M' \in \mathcal{S}' : M' \text{ is an independent set}\}$,

:: let $\mathcal{S}'_{two} = \{M' \in \mathcal{S}' : M' \text{ is a } q\text{-star } m\text{-full set for some } q \in \{1, \dots, l - p' + 1\}$,

$M' \text{ is not an independent set}\}$,

:: $\mathcal{Q}(i, H) := \{\{v_i\} \cup H_0 \cup M' : M' \in \mathcal{S}'_{one}\}$,

:: $\mathcal{S}_i := \mathcal{S}_i \cup \mathcal{Q}(i, H)$ (and say that each member of $\mathcal{Q}(i, H)$ is *unmarked*)

:: $\mathcal{T}(i, H) := \{\{v_i\} \cup H_0 \cup M' : M' \in \mathcal{S}'_{two}$

whose q stars are of centers say c'_1, \dots, c'_q for some $q \in \{1, \dots, l - p' + 1\}\}$

:: $\mathcal{S}_i := \mathcal{S}_i \cup \mathcal{T}(i, H)$

(and say that each member $H' = \{v_i\} \cup H_0 \cup M' \in \mathcal{T}(i, H)$ is $p' + q$ -marked

and that the markers of H' are

$u_j(H') := u_j(H)$, for $j = 1, \dots, p'$, and $u_{p'+j}(H') := c'_j$, for $j = 1, \dots, q$).

(2.2) If H is p -marked for some $p \in \{2, \dots, l + 1\}$,

and if v_i is either of Type 1 being adjacent to at least two markers of H or of Type 5,

then

:: compute the family \mathcal{R} of all maximal independent sets of $G[H^* \setminus N(v_i)]$

:: $\mathcal{R}(i, H) := \{\{v_i\} \cup H_0 \cup R : R \in \mathcal{R}\}$

:: $\mathcal{S}_i := \mathcal{S}_i \cup \mathcal{R}(i, H)$ (and say that each member of $\mathcal{R}(i, H)$ is *unmarked*)

Step 3 [Addition of 1-marked members to \mathcal{S}_i]

For each induced star (c, L) of G_i such that $|L| \leq m$,

$H := L \cup A_{G_i}(c, L)$,

$\mathcal{S}_i := \mathcal{S}_i \cup \{H\}$ (and say that H is 1-marked: the marker of H is $u_1(H) := c$)
 end
 $\mathcal{S} := \mathcal{S}_i$
 end.

Let us prove the following lemmas.

Note: Statement (i') of the subsequent Lemma 7 ensures that Step 2 of Algorithm Beta(l, m) is well defined, in fact $G[H^* \setminus N(v_i)]$ needs to be $(l - p')$ $K_2 + Y_{m,m}$ -free in order to execute Algorithm Beta($l - p', m$) for $G[H^* \setminus N(v_i)]$.

Lemma 7 Referring to Algorithm Beta(l, m) for any $l \geq 1$:

- (i) every 1-marked member H of \mathcal{S}_i , for $i = 1, \dots, n$, is a 1-star set whose star is of center $u_1(H)$ and of leaves a subset of H_0 ;
- (ii) every p -marked member H for $p \in \{2, \dots, l + 1\}$ of \mathcal{S}_i , for $i = 1, \dots, n$, is a p -star m -full set whose stars respectively are of center $u_1(H), \dots, u_p(H)$ and of leaves a subset of H_0 ;
- (iii) referring to Step (2.1) of Loop i of Algorithm Beta(l, m), for $i = 1, \dots, n$, $G[H^* \setminus N(v_i)]$ is $(l - p')$ $K_2 + Y_{m,m}$ -free;
- (iv) referring to Step (2.2) of Loop i of Algorithm Beta(l, m), for $i = 1, \dots, n$, $G[H^* \setminus N(v_i)]$ is IK_2 -free;
- (v) every unmarked member of \mathcal{S}_i , for $i = 1, \dots, n$, is an independent set;
- (vi) the family \mathcal{S} produced by Algorithm Beta(l, m) can be partitioned into $\{\mathcal{S}_{one}, \mathcal{S}_{two}, \mathcal{S}_{three}\}$, where: $\mathcal{S}_{one} = \{I \in \mathcal{S} : I \text{ is an independent set}\}$, $\mathcal{S}_{two} = \{I \in \mathcal{S} : I \text{ is a } p\text{-star } m\text{-full set for some } p \in \{1, \dots, l + 1\}, I \text{ is not an independent set}\}$, $\mathcal{S}_{three} = \{I \in \mathcal{S} : I \text{ is a } p\text{-star set for some } p \in \{1, \dots, l + 1\}, I \text{ is not an independent set, } I \text{ is not a } p\text{-star set } m\text{-full for } p \in \{1, \dots, l + 1\}\}$.

Proof The proof is by induction on l . For $l = 1$ the proof is that of Lemma 4. Then let us assume that the lemma holds for $l - 1$ and let us show that it holds for l .

Proof of (i). The proof is similar to that of Lemma 4 (i).

Proof of (ii). Each p -marked member for $p \in \{2, \dots, l + 1\}$ is created in Step (2.1) of some loop with respect to a p' -marked member, with $p' < p$, and then it is possibly extended in Step (1.2) of the successive loops. Then, by the inductive assumption on l (i.e., by Statement (iv) concerning Algorithm Beta($l - p', m$)), by definition of Step (2.1) and by definition of $u_1(H), \dots, u_p(H)$, each p -marked member satisfies Statement (ii) at the moment of its creation. Furthermore each p -marked member satisfies Statement (ii) after its possible extensions by definition of a vertex of Type 1 being adjacent to at most one marker of H and of Type 3 (let us observe that each p -marked member H keeps its markers $u_1(H), \dots, u_p(H)$ along its possible extensions).

Proof of (i'). At Step (2.1) of Loop i of Algorithm Beta(l, m), for $i = 1, \dots, n$, by construction $H^* \setminus N(v_i)$ is contained in the anti-neighborhood of $\{v_i\} \cup \{u_1(H), \dots, u_{p'}(H)\} \cup (N(u_1) \cap H_0) \cup \dots \cup (N(u_{p'}) \cap H_0)$, for some

$p' \in \{1, \dots, l\}$, which is a collection of p' stars of centers $u_1(H), \dots, u_{p'}(H)$ having mutually a co-join (thus containing an induced $p'K_2$) by Statements (i)–(ii) and since v_i is of Type 4. Then Statement (i') follows since G is $lK_2 + Y_{m,m}$ -free.

Proof of (ii'). At Step (2.2) of Loop i of Algorithm Beta(l, m), for $i = 1, \dots, n$, H is p -marked for some $p \in \{2, \dots, l + 1\}$ and v_i is either of Type 1 being adjacent to at least two markers of H or of Type 5, say without loss of generality, v_i is adjacent to $u_1(H)$ and $u_2(H)$. Therefore, by Statement (ii), v_i and the two stars of H of center $u_1(H)$ and $u_2(H)$ induce a subgraph of G_i that contains an induced $Y_{m,m}$. Then Statement (ii') follows since G is $lK_2 + Y_{m,m}$ -free.

Proof of (iii). The proof is similar to that of Lemma 4 (iii).

Proof of (iv). It follows by Statements (i)–(iii) and by definition of Algorithm Beta (l, m). □

Lemma 8 *Let \mathcal{S} be the family of subsets of $V(G)$ produced by Algorithm Beta(l, m) for any $l \geq 1$. Then for any maximal independent set I of G , there is a member $H \in \mathcal{S}$ such that $I \subseteq H$.*

Proof The proof is by induction on l . For $l = 1$ the proof is that of Lemma 5. Then let us assume that the lemma holds for $l - 1$ and let us show that it holds for l .

The first part of this proof is similar to the first part of the proof of Lemma 2.

Case 1. H is unmarked.

This case is similar to Case 1 of Lemma 2, with Lemma 7 instead of Lemma 1, and with Step 3 instead of Step 2.

Case 2. H is p -marked for some $p \in \{1, \dots, l + 1\}$.

Case 2.1. $p \in \{1, \dots, l\}$.

By Lemma 7 (i)–(ii) H is a p -star set of G whose stars are of center $u_1(H), \dots, u_p(H)$ and of leaves a subset of H_0 . If v_i is of Type 1 with respect to H being adjacent to at most one marker of H or of Type 3 with respect to H , then $I \subseteq H \cup \{v_i\} \in \mathcal{S}_i$ (Step 1). If v_i is of Type 2 with respect to H , then there exists a vertex $u \in N_{G_i}(v_i)$ that has no neighbors in $I \setminus \{v_i\}$, that is, I is contained in the set $\{v_i\} \cup A_{G_i}(u, v_i) \in \mathcal{S}_i$ (Step 3).

Let us consider the case in which v_i is either of Type 1 with respect to H being adjacent to at least two markers of H or of Type 5 with respect to H (so that $p \geq 2$).

Then I is contained in some member of the family $\mathcal{R}(i, H)$ generated by Step (2.2) of Loop i of Algorithm Beta(l, m).

Let us consider the case in which v_i is of Type 4 with respect to H .

Since I is a maximal independent set of G_i , I is a maximal independent set of $G[H \cup \{v_i\}]$ as well, i.e., $I \setminus \{v_i\}$ is a maximal independent set of $G[H \setminus N(v_i)]$. In particular, since v is of Type 4, one has that v_i is adjacent to exactly one marker of H and that $H_0 \subset I \setminus \{v_i\}$.

If H is not m -full, i.e., $|N(u_j(H)) \cap H_0| < m$ for some $j \in \{1, \dots, p\}$, then I is contained in the set $\{v_i\} \cup (N(u_j(H)) \cap H_0) \cup A_{G_i}(u_j(H), \{v_i\} \cup (N(u_j(H)) \cap H_0)) \in \mathcal{S}_i$ (Step 3).

If H is m -full, i.e., $|N(u_j(H)) \cap H_0| \geq m$ for $j \in \{1, \dots, p\}$, then let us consider the following argument.

Let us write $I = \{v_i\} \cup H_0 \cup I'$, where $I' = (I \setminus \{v_i\}) \cap (H^* \setminus N(v_i))$. Since $I \setminus \{v_i\}$ is a maximal independent set of $G[H \setminus N(v_i)]$, I' is a maximal independent set of $G[H^* \setminus N(v_i)]$. Now, according to Step (2.1) of Algorithm Beta(l, m) and according to Lemma 7 (iv) by the inductive assumption on l , let $S' = S'_{one} \cup S'_{two} \cup S'_{three}$ be the family of subsets of $V(G[H^* \setminus N(v_i)])$ produced by Algorithm Beta($l - p, m$) executed for $G[H^* \setminus N(v_i)]$. Therefore, by the present lemma and by the inductive assumption on l , there exists an $M' \in S'$ such that $I' \subseteq M'$, and by Lemma 7 (iv) and by the inductive assumption on l , one of the following three subcases occurs.

Assume that $M' \in S'_{one}$. Then $\{v_i\} \cup H_0 \cup M'$ belongs to the family $\mathcal{Q}(i, H)$ generated by Step (2.1) of Loop i of Algorithm Beta(l, m), i.e., I is contained in $\{v_i\} \cup H_0 \cup M' \in S_i$.

Assume that $M' \in S'_{two}$. Then $\{v_i\} \cup H_0 \cup M'$ belongs to the family $\mathcal{T}(i, H)$ generated by Step (2.1) of Loop i of Algorithm Beta(l, m), i.e., I is contained in $\{v_i\} \cup H_0 \cup M' \in S_i$.

Assume that $M' \in S'_{three}$. Therefore, M' is a p' -star set of $G[H^* \setminus N(v_i)]$ for some $p' \in \{1, \dots, l - p + 1\}$, whose at least one star say (c', L') is such that $|L'| < m$ and is clearly a star of G_i as well. Then I is contained in the set $L' \cup A_{G_i}(c', L') \in S_i$ (Step 3).

Case 2.2. $p = l + 1$.

By Lemma 7 (ii) H is a p -star m -full set of G whose stars are of center $u_1(H), \dots, u_p(H)$ and of leaves a subset of H_0 . If v_i is of Type 1 is either of Type 1 with respect to H being adjacent to at most one marker of H or of Type 3 with respect to H , then $I \subseteq H \cup \{v_i\} \in S_i$ (Step 1). If v_i is of Type 2 with respect to H , then there exists a vertex $u \in N_{G_i}(v_i)$ that has no neighbors in $I \setminus \{v_i\}$, that is, I is contained in the set $\{v_i\} \cup A_{G_i}(u, v_i) \in S_i$ (Step 3).

Let us consider the case in which v_i is either of Type 1 with respect to H being adjacent to at least two markers of H or of Type 5 with respect to H (so that $p \geq 2$).

Then I is contained in some member of the family $\mathcal{R}(i, H)$ generated by Step (2.2) of Loop i of Algorithm Beta(l, m).

Let us consider the case in which v_i is of Type 4 with respect to H .

Let U_j , for $j = 1, \dots, p$, denote the set of neighbors of $u_j(H)$ in H_0 ; then by the above $|U_j| \geq m$.

Without loss of generality, by symmetry, assume that v_i is adjacent to $u_1(H)$ and is non-adjacent to $u_2(H), \dots, u_p(H)$.

Since v_i is of Type 4, v_i is adjacent to a vertex say h of H^* . Let us write $T = N(h) \cap (I \setminus \{v_i\})$. Note that, by Lemma 7 (ii), $\{u_1(H)\} \cup U_1$ has a co-join to $\{h\} \cup T$.

Assume that $T = \emptyset$. Then I is contained in the set $\{v_i\} \cup A_{G_i}(h, v_i) \in S_i$ (Step 3).

Assume that $T \neq \emptyset$. Therefore $|T| < m$, else vertices of $T, \{h, v_i, u_1(H)\}, U_1$ would induce a $Y_{m,m}$, that is, any induced IK_2 of the remaining stars of H of center $u_2(H), \dots, u_p(H)$ (recall that $p = l + 1$) and such an induced $Y_{m,m}$ would induce a $IK_2 + Y_{m,m}$, that is not possible since G is $IK_2 + Y_{m,m}$ -free. Then I is contained in the set $\{v_i\} \cup T \cup A_{G_i}(h, \{v_i\} \cup T) \in S_i$ (Step 3). □

The next lemma is the homologous of Lemma 6. However let us omit a detailed analysis on polynomial time bounds since that would comport a (not difficult but) recursive analysis with long expressions.

Lemma 9 *Referring to Algorithm Beta(l,m):*

- (i) for $i = 1, \dots, n$, and for any p' -marked member H for $p' \in \{1, \dots, l\}$, the family $\mathcal{Q}(i, H) \cup \mathcal{T}(i, H)$ contains polynomially many members and can be computed in polynomial time;
- (ii) for $i = 1, \dots, n$, and for any p -marked member H for $p \in \{2, \dots, l + 1\}$, the family $\mathcal{R}(i, H)$ contains polynomially many members and can be computed in polynomial time;
- (iii) Algorithm Beta(l,m) runs in polynomial time and computes a family \mathcal{S} of polynomial size.

Proof For $l = 1$ the proof is that of Lemma 6. Then let us assume that the lemma holds for $l - 1$ and let us show that it holds for l .

Proof of (i): It follows by definition of Step (2.1), i.e., the execution of Algorithm Beta($l - p', m$), and by the induction assumption on l .

Proof of (ii): It follows by definition of Step (2.2); in fact, by Lemma 4 (ii'), $G[H^* \setminus N(v_i)]$ is IK_2 -free; therefore $G[H \setminus N(v_i)]$ is IK_2 -free by definition of H_0 ; then $G[H \setminus N(v_i)]$ contains polynomially many maximal independent sets by Theorem 4 and by Theorem 5, i.e., the family $\mathcal{R}(i, H)$ contains polynomially many members.

Proof of (iii):

Concerning 1-marked members of \mathcal{S} . The total number of new 1-marked members added to \mathcal{S} by each Step 3 of each loop of Algorithm Beta(l, m) is at most n^{m+2} (that is at most n^{m+1} for each loop); in particular, since each 1-marked member can be created/extended in $O(n^2)$ time (in particular in order to determine the connected components) and since the loops are n , all the marked members can be computed in $O(n^{m+4})$ time.

Concerning p -marked members of \mathcal{S} for $p \in \{2, \dots, l + 1\}$. Let us observe that each p -marked member for $p \in \{2, \dots, l + 1\}$ is created in Step 2 of some loop with respect to a p' -marked member for $p' \in \{1, \dots, l\}$, with $p' < p$, and then it is possibly extended in Step (1.2) of the successive loops. Then let us prove that Algorithm Beta (l, m) produces polynomially many p -marked members for $p \in \{2, \dots, l + 1\}$ by induction on p .

For $p = 2$: By statement (i), for $i = 1, \dots, n$, and for any 1-marked member H , the family $\mathcal{T}(i, H)$ of 2-marked members contains polynomially many members and can be computed in polynomial time. Therefore, since the total number of 1-marked members is bounded by a polynomial (by the above), the total number of 2-marked members from families $\mathcal{T}(i, H)$ (for $i = 1, \dots, n$, and for any 1-marked member H) is bounded by a polynomial, and all such members can be computed in polynomial time; in particular, since each 2-marked member can be extended in $O(n)$ time and since the loops are n , all 2-marked members can be computed in polynomial time.

For $p > 2$: By statement (i), for $i = 1, \dots, n$, and for any p' -marked member H with $p' < p$, the family $\mathcal{T}(i, H)$ of p -marked members contains polynomially many

members and can be computed in polynomial time. Therefore, since the total number of p' -marked members with $p' < p$ is bounded by a polynomial *by the inductive assumption on p* , the total number of p -marked members from families $\mathcal{T}(i, H)$ (for $i = 1, \dots, n$, and for any p' -marked member H with $p' < p$) is bounded by a polynomial, and all such members can be computed in polynomial time; in particular, since each p -marked member can be extended in $O(n)$ time and since the loops are n , all p -marked members can be computed in polynomial time.

Concerning unmarked members of \mathcal{S} . Let us observe that each unmarked member is created with respect to some p' -marked member for $p' \in \{1, \dots, l\}$ in Step (2.1) or with respect to some p -marked member for $p \in \{2, \dots, l + 1\}$ in Step (2.2), and then it is possibly extended in Step 1 of the successive loops (except for the unmarked member which is created as the empty set in the initialization step and is possibly extended in Step 1 of the successive loops).

By statement (i), for $i = 1, \dots, n$, and for any p' -marked member H , the family $\mathcal{Q}(i, H)$ of unmarked members contains polynomially many elements and can be computed in polynomial time. Therefore, since the total number of p' -marked members is bounded by a polynomial (by the above), the total number of unmarked numbers from families $\mathcal{Q}(i, H)$ (for $i = 1, \dots, n$, and for any p' -marked member H) is bounded by a polynomial, and all such members can be computed in polynomial time; in particular, since each unmarked member can be extended in polynomial time and since the loops are n , all such unmarked members can be computed in polynomial time.

By statement (ii), for $i = 1, \dots, n$, and for any p -marked member H , the family $\mathcal{R}(i, H)$ of unmarked members contains polynomially many elements and can be computed in polynomial time. Therefore, since the loops of the algorithm are n , and since the total number of p -marked members is bounded by a polynomial (by the above), the total numbers of unmarked members from families $\mathcal{R}(i, H)$ (for $i = 1, \dots, n$, and for any p -marked member H) is bounded by a polynomial, and all such members can be computed in polynomial time; in particular, since each unmarked member can be extended in polynomial time and since the loops are n , all such unmarked members can be computed in polynomial time.

Summarizing: the total number of members of the family \mathcal{S} produced by Algorithm Beta(l, m) is bounded by a polynomial, and all such members can be computed in polynomial time, which is also the time bound of Algorithm Beta(l, m). \square

Then, let us consider the following algorithm to solve WIS for $IK_2 + \mathcal{Y}$ -free graphs for any $l \geq 1$, where \mathcal{Y} is any graph family containing an induced subgraph of $Y_{m,m}$.

Algorithm WIS(l, m)

Input: A $lK_2 + \mathcal{Y}$ -free graph G .

Output: A maximum weight independent set of G .

PHASE 1. Execute Algorithm Beta(l, m) for G . Let \mathcal{S} be the family of subsets of G produced in this way.

PHASE 2. For each $H \in \mathcal{S}$, compute a maximum weight independent set of $G[H]$. Then choose a best solution, i.e., one of maximum weight.

Theorem 9 *Algorithm WIS*(l, m) *is correct and can be executed in polynomial time whenever WIS can be solved for* \mathcal{Y} -free graphs *in polynomial time.*

Proof The proof is by induction on l . For $l = 1$ the proof is that of Theorem 7. Then let us assume that the theorem holds for $l - 1$ and let us show that it holds for l .

Correctness. By Lemma 8, Algorithm WIS(l, m) is correct.

Complexity.

Concerning PHASE 1: By Lemma 9, PHASE 1. can be executed in polynomial time.

Concerning PHASE 2: By Lemma 9, the family \mathcal{S} contains polynomially many members. On the other hand, by Lemma 7 (ii) and by Observation 1, for each $H \in \mathcal{S}$ only two cases may occur: either H is an independent set (and then the problem is trivial), or each non-trivial component K of $G[H]$ is contained in the anti-neighborhood of one induced edge of G (easily detectable), i.e., K is $(l - 1)K_2 + \mathcal{Y}$ -free since G is $lK_2 + \mathcal{Y}$ -free. Therefore, *by the inductive assumption on l* , PHASE 2 can be executed in polynomial time whenever WIS can be solved for \mathcal{Y} -free graphs in polynomial time.

Then Algorithm WIS(l, m) can be executed in polynomial time whenever WIS can be solved for \mathcal{Y} -free graphs in polynomial time. □

Then one obtains the following result.

Theorem 10 *For any constant m and for any graph family \mathcal{Y} containing an induced subgraph of $Y_{m,m}$, if WIS can be solved for \mathcal{Y} -free graphs in polynomial time, then WIS can be solved for $lK_2 + \mathcal{Y}$ -free graphs in polynomial time for any constant l . □*

Acknowledgements Please I would like to thank the referees, concerning a previous submission, for their helpful and valuable comments (in particular for having detected an error in the first version). Additionally I am grateful and indebted to Prof. Andreas Brandstädt and to Prof. Vadim V. Lozin for helpful discussions on the topic and for their helpful remarks. Finally I would like to witness that just try to pray a lot and am not able to do anything without that - ad laudem Domini.

Funding Open access funding provided by Università degli Studi G. D'Annunzio Chieti Pescara within the CRUI-CARE Agreement.

Declarations

Conflict of interest The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alekseev, V.E.: On the local restriction effect on the complexity of finding the graph independence number. In: *Combinatorial-algebraic Methods in Applied Mathematics*, pp. 3–13. Gorkiy University Press, Gorkiy (1983). **(in Russian)**
2. Alekseev, V.E.: On the number of maximal independent sets in graphs from hereditary classes. In: *Combinatorial-algebraic Methods in Discrete Optimization*, pp. 5–8. Gorkiy University Press, Gorkiy (1991). **(in Russian)**
3. Alekseev, V.E.: A polynomial algorithm for finding largest independent sets in graphs without forks, *Discrete analysis and operations research ser. 1*, 6 (1999) 3–19 (in Russian). *Discrete Appl. Math.* **135**, 3–16 (2004)
4. Alekseev, V.E.: On easy and hard hereditary classes of graphs with respect to the independent set problem. *Discret. Appl. Math.* **132**, 17–26 (2003)
5. Bacsó, G., Lokshantov, D., Marx, D., Pilipczuk, M., Tuza, Z., van Leeuwen, E.J.: Subexponential-time algorithms for maximum independent set in P_t -free and broom-free graphs. *Algorithmica* **81**, 421–438 (2019)
6. Brandstädt, A., Le, V.B., Spinrad, J.P.: *Graph Classes: A Survey*, SIAM Monographs on Discrete Math. Appl., vol. 3. SIAM, Philadelphia (1999)
7. Brandstädt, A., Lozin, V.V., Mosca, R.: Independent sets of maximum weight in apple-free graphs. *SIAM J. Discrete Math.* **24**, 239–254 (2010)
8. Brandstädt, A., Klembt, T., Lozin, V.V., Mosca, R.: Independent sets of maximum weight in apple-free graphs, *ISAAC 2008. LNCS 5369*, 848–858 (2008)
9. Brandstädt, A., Mosca, R.: Maximum weight independent set for l -claw-free graphs in polynomial time. *Discret. Appl. Math.* **237**, 57–64 (2018)
10. Brause, C.: A subexponential-time algorithm for the Maximum Independent Set Problem in P_t -free graphs. *Discret. Appl. Math.* **231**, 113–118 (2017)
11. Corneil, D.G., Perl, Y., Stewart, L.K.: A linear recognition algorithm for cographs. *SIAM J. Comput.* **14**, 926–934 (1985)
12. Dabrowski, K.K., Lozin, V.V., de Werra, D., Zamaraev, V.: Combinatorics and algorithms for augmenting graphs. *Graphs Comb.* **32**, 1339–1352 (2016)
13. Faenza, Y., Oriolo, G., Stauffer, G.: An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$ -algorithm for the weighted stable set problem. In: *SODA*, pp. 630–646 (2011)
14. Farber, M.: On diameters and radii of bridged graphs. *Discrete Math.* **73**, 249–260 (1989)
15. Farber, M., Hujter, M., Tuza, Z.: An upper bound on the number of cliques in a graph. *Networks* **23**, 75–83 (1993)
16. Fricke, G.H., Hedetniemi, S.T., Jacobs, D.P.: Independence and irredundance in k -regular graphs. *Ars Comb.* **49**, 271–279 (1998)

17. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. *Theor. Comput. Sci.* **1**, 237–267 (1976)
18. Garey, M.R., Johnson, D.S.: The rectilinear Steiner tree problem is NP-complete. *SIAM J. Appl. Math.* **32**, 826–834 (1977)
19. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco (1979)
20. Gartland, P., Lokshtanov, D.: Independent set on P_k -free graphs in quasi-polynomial time. CoRR abs arXiv:2005.00690
21. Grötschel, M., Lovász, L., Schrijver, A.: Polynomial algorithms for perfect graphs. *Ann. Discrete Math.* **21**, 325–356 (1984)
22. Grzesik, A., Klimošová, T., Pilipczuk, M., Pilipczuk, M.: Polynomial-time algorithm for maximum weight independent set on P_6 -free graphs. In: Chan, T.M. (e.), *Proceedings of the 18 Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6–9, 2019*, pp. 1257–1271. SIAM (2019)
23. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) *Complexity of Computer Computations*, pp. 85–103. Plenum Press, New York (1972)
24. Lokshtanov, D., Pilipczuk, M., van Leeuwen, E. J.: Independence and efficient domination on P_6 -free graphs. *Proc. SODA 2016*, pages 1784–1803. *ACM Trans. Algorithms* **14**(1), 3:1–3:30 (2018)
25. Lokshtanov, D., Vatschelle, M., Villanger, Y.: Independent sets in P_5 -free graphs in polynomial time. In: *Proc. SODA*, pp. 570–581 (2014)
26. Lozin, V.V.: From matchings to independent sets. *Discret. Appl. Math.* **231**, 4–14 (2017)
27. Lozin, V.V., Milanič, M.: A polynomial algorithm to find an independent set of maximum weight in a fork-free graph, RUTCOR Research Report, Rutgers University, 30-2005. *J. Discrete Algorithms* **6**, 595–604 (2008)
28. Lozin, V.V., Mosca, R.: Maximum independent sets in planar graphs. RUTCOR Research Report RRR 40-2004
29. Lozin, V.V., Mosca, R.: Independent sets in extensions of $2K_2$ -free graphs. *Discret. Appl. Math.* **146**, 74–80 (2005)
30. Lozin, V.V., Mosca, R.: Maximum regular subgraphs in $2P_3$ -free graphs. *Theor. Comput. Sci.* **460**, 26–33 (2012)
31. Lozin, V.V., Mosca, R., Purcell, C.: Sparse regular induced subgraphs in $2P_3$ -free graphs. *Discrete Optim.* **10**, 304–309 (2013)
32. Minty, G.J.: On maximal independent sets of vertices in claw-free graphs. *J. Comb. Theory Ser. B* **28**, 284–304 (1980)
33. Mosca, R.: Maximum weight independent set in $(P_6, \text{co-banner})$ -free graphs. *Inf. Process. Lett.* **113**, 89–93 (2013)
34. Mosca, R.: A sufficient condition to extend polynomial results for the maximum independent set problem. *Discrete Appl. Math.* **216**, 281–289 (2017)
35. Murphy, O.J.: Computing independent sets in graphs with large girth. *Discrete Appl. Math.* **35**, 167–170 (1992)
36. Nakamura, D., Tamura, A.: A revision of Minty’s algorithm for finding a maximum weight stable set in a claw-free graph. *J. Oper. Res. Soc. Jpn.* **44**, 194–204 (2001)
37. Nobili, P., Sassano, A.: An $O(n^2 \log(n))$ algorithm for the weighted stable set problem in claw-free graphs. *Math. Progr.* **186**, 409–437 (2021)
38. Nobili, P., Sassano, A.: An algorithm for the weighted stable set problem in claw, net-free graphs with $\alpha(G) \geq 4$. *Discrete Optim.* **19**, 63–78 (2016)
39. Nobili, P., Sassano, A.: An $\mathcal{O}(n^2 \log n)$ algorithm for the weighted stable set problem in claw-free graphs with $\alpha(G) \leq 3$. *Math. Progr.* **164**, 157–165 (2017)
40. Pilipczuk, M., Pilipczuk, M., Rzazewski, P.: Quasi-polynomial-time algorithm for independent set in P_t -free graphs via shrinking the space of induced paths. *SOSA 204–209* (2021). <https://doi.org/10.1137/1.9781611976472.23>
41. Poljak, S.: A note on stable sets and colorings of graphs. *Commun. Math. Univ. Carolinae* **15**, 307–309 (1974)
42. Prisner, E., *Graphs with few cliques*, *Graph Theory, Combinatorics and Algorithms*, vol. 1, 2 (Kalamazoo, MI, 1992), pp. 945–956. Wiley-Interscience Publishers, Wiley, New York (1995)
43. Sbihi, N.: Algorithmes de recherche d’un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Math.* **29**, 53–76 (1980)

44. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all maximal independent sets. *SIAM J. Comput.* **6**, 505–517 (1977)

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