Maximum Weight Independent Set for ℓClaw-Free Graphs in Polynomial Time

Andreas Brandstädt[1] Raffaele Mosca[2]

October 27, 2017

[1] Institut für Informatik, Universität Rostock, D-18051 Rostock, Germany.
andreas.brandstaedt@uni-rostock.de

[2] Dipartimento di Economia, Universitá degli Studi "G. D'Annunzio", Pescara 65121, Italy.
r.mosca@unich.it

Abstract

For a finite, simple and undirected graph G with vertex weight function, the Maximum Weight Independent Set (MWIS) problem asks for an independent vertex set of G with maximum weight. MWIS is a well-known NP-hard problem of fundamental importance. For many classes of graphs, MWIS can be solved in polynomial time; famous examples are bipartite graphs, perfect graphs and claw-free graphs. In 1980, the first two polynomial-time algorithms for claw-free graphs were independently found by Minty for MWIS and by Sbihi for the unweighted case.

In this paper, using a dynamic programming approach (inspired by Farber's result about $2K_2$-free graphs), we show that for any fixed ℓ, MWIS can be solved in polynomial time for ℓclaw-free graphs. This solves the open case for MWIS on (P_3+claw)-free graphs and ($2P_2$+claw)-free graphs extends known results for claw-free graphs, ℓK_2-free graphs, (K_2+claw)-free graphs, and ℓP_3-free graphs.

Keywords: Maximum Weight Independent Set problem; ℓclaw-free graphs; polynomial time

1 Introduction

For a finite, undirected and simple graph $G = (V, E)$ with $|V| = n$, the *Maximum Independent Set* (*MIS*) problem asks for an independent vertex set of maximum cardinality in G. MIS is well known to be NP-hard (problem [GT20] in [17]) and hard to approximate. If additionally, the input graph has a vertex weight function w, the *Maximum Weight Independent Set* (*MWIS*) problem asks for an independent vertex set of maximum weight in G. MWIS is one of the most investigated and most important algorithmic problem on graphs because of its many applications in various fields of research such as computer science and operations research.

More precisely, for a given graph $G = (V, E)$ with a vertex weight function w and for $U \subseteq V$, $w(U) := \Sigma_{v \in U} w(v)$ is the *weight of U*. An *independent set* (also called *stable set*) of a graph G is a subset of V of pairwise nonadjacent vertices in G. An independent set of G is *maximal* if it is not properly contained in any other independent set of G. Let

$$\alpha_w(G) := \max\{w(I) : I \text{ is an independent set in } G\}.$$

If all vertices v have the same weight $w(v) = 1$ (the *unweighted case*) then $\alpha_w(G) = \alpha(G)$ and we deal with the MIS problem.

It is well known that MWIS is solvable in polynomial time for many special cases; famous examples are bipartite graphs, perfect graphs [18] and claw-free graphs [25], and in many other cases, the complexity of MWIS is still an open question. Before discussing this in detail, we need some more notions (for any missing notation or reference let us refer to [6]).

1.1 Basic Notions and Results

For $U \subseteq V$ let $G[U]$ denote the subgraph of G induced by U. For a given graph H, a graph G is *H-free* if none of its induced subgraphs is isomorphic to H; in particular, H is called a *forbidden induced subgraph of G*.

For two graphs G and F, $G+F$ denotes the *disjoint union* of G and F; in particular, $2G = G + G$ and in general, for $\ell \geq 2$, ℓG denotes the disjoint union of ℓ copies of G.

For a subset $U \subseteq V$, the *open neighborhood* of U in G is $N_G(U) = \{v \in V \setminus U : v \text{ is adjacent to some } u \in U\}$ and the *closed neighborhood* of U in G is $N_G[U] = U \cup N_G(U)$. The *anti-neighborhood* of U in G is $A_G(U) = V \setminus N_G[U]$, also denoted as $\overline{N_G[U]}$. For an induced subgraph H of G with vertex set $V(H)$, let $A_G(H) := A_G(V(H))$. If $U = \{u_1, \ldots, u_k\}$, then let us simply write $N_G(u_1, \ldots, u_k)$ instead of $N_G(U)$, and $A_G(u_1, \ldots, u_k)$ instead of $A_G(U)$.

For a vertex $v \in V$ and for a subset $U \subset V$ with $v \notin U$, let us say that v *contacts* U if v is adjacent to some vertex of U. A *component of G* is a maximal connected subgraph of G.

A *chordless path* P_k, $k \geq 1$, has vertices $v_1, v_2, \ldots, v_k$ and exactly the edges $v_j v_{j+1}$ for $1 \leq j < k$. A *chordless cycle* C_k, $k \geq 4$, has vertices $v_1, v_2, \ldots, v_k$ and exactly the edges $v_j v_{j+1}$ for $1 \leq j < k$ and $v_k v_1$. K_n is a complete graph of n vertices. $K_{1,i}$ has $i+1$ vertices such that one of them (called the *center* of $K_{1,i}$) is adjacent to all others, and the remaining i vertices (called the *leaves* of $K_{1,i}$) form an independent set. $K_{1,3}$ is also called *claw*.

An *apple* is formed by a C_k, $k \geq 4$, plus one vertex adjacent to exactly one vertex of the C_k.

For indices $i, j, k \geq 0$, let $S_{i,j,k}$ denote the graph with vertices $u, x_1, \ldots, x_i, y_1, \ldots, y_j, z_1, \ldots, z_k$ such that the subgraph induced by $u, x_1, \ldots, x_i$ forms a P_{i+1} $(u, x_1, \ldots, x_i)$, the subgraph induced by $u, y_1, \ldots, y_j$ forms a P_{j+1} $(u, y_1, \ldots, y_j)$, and the subgraph induced by $u, z_1, \ldots, z_k$ forms a P_{k+1} $(u, z_1, \ldots, z_k)$, and there are no other edges in $S_{i,j,k}$. Thus, $S_{1,1,1}$ is isomorphic to claw, $S_{1,1,2}$ is called *fork*, and P_k is isomorphic to e.g. $S_{0,0,k-1}$.

As already mentioned, MWIS is NP-hard and remains NP-hard under various restrictions, such as for triangle-free graphs [30] and more generally for graphs without chordless cycle of given length [26], for $K_{1,4}$-free graphs [25], for cubic graphs [16] and more generally for k-regular graphs [14], and for planar graphs [15].

MWIS for claw-free graphs was first solved in polynomial time by Minty [25] in 1980 (and independently for MIS by Sbihi), then revisited by Nakamura and Tamura [27], and recently improved by Faenza, Oriolo, and Stauffer [10, 11], and by Nobili and Sassano [28, 29] with the best known time bound in [29].

Theorem 1 [29] *For claw-free graphs, the MWIS problem can be solved in time $\mathcal{O}(n^2 \log n)$.*

Other polynomial-time results are obtained for the more general cases of fork-free (i.e., $S_{1,1,2}$-free) graphs (in [3] for MIS, and in [22] for MWIS) and for apple-free graphs [7, 8]. Furthermore MWIS can be solved in polynomial time for P_5-free graphs [20] and even for P_6-free graphs as recently proved in [19].

A dynamic programming approach leads to polynomial time for MWIS if there are polynomially many subgraphs containing any maximal independent set such that the corresponding maximal independent set in the subgraph can be found in polynomial time.

Examples are $2K_2$-free graphs [12] having $\mathcal{O}(n^2)$ maximal independent sets, and more generally ℓK_2-free graphs for any fixed ℓ (by combining an algorithm generating all maximal independent sets of a graph [33] and a polynomial upper bound on the number of maximal independent sets in ℓK_2-free graphs [2, 5, 13, 31]), (K_2+claw)-free graphs [23], $2P_3$-free graphs [24] and more generally, ℓP_3-free graphs [21].

Obviously, for every graph G the following holds:

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

Thus, for any graph G, MWIS can be reduced to the same problem for the anti-neighborhoods of all vertices of G. Then we have:

Proposition 1 *For any graph F, if M(W)IS can be solved for F-free graphs in polynomial time then M(W)IS can be solved for $(K_1 + F)$-free graphs in polynomial time.*

The following result of Alekseev [1] (mentioned e.g. in [4]) shows the fundamental importance of $S_{i,j,k}$ for the complexity of MWIS. Let us say that a graph is of *type T* if it is a graph $S_{i,j,k}$ for some indices i, j, k.

Theorem 2 [1] *Let M be a finite set of forbidden induced subgraphs. If M does not contain a graph every connected component of which is of type T, then the M(W)IS problem is* NP*-hard for the class of M-free graphs.*

A basic example is the class of $K_{1,4}$-free graphs as mentioned above. Unless P = NP, Alekseev's result implies that for any graph F, if M(W)IS is polynomial time solvable for F-free graphs, then each connected component of F is of type T. By Proposition 1, for any graph F, if M(W)IS can be solved in polynomial time for F-free graphs then for any fixed ℓ, M(W)IS can be solved in polynomial time for $(\ell K_1 + F)$-free graphs.

As already mentioned above, for any fixed ℓ, M(W)IS can be solved in polynomial time for ℓK_2-free graphs, for fork-free graphs, for (K_2+claw)-free graphs, for ℓP_3-free graphs, and for P_5-free graphs. On the other hand the F-free graph classes defined by minimal forbidden graphs F of type T for which the complexity of M(W)IS seems to be open are the following:

> P_7-free graphs, $S_{1,1,3}$-free graphs, $S_{1,2,2}$-free graphs, $(K_2 + P_4)$-free graphs, (P_3+claw)-free graphs, ($2P_2$+claw)-free graphs.

In this paper, using the dynamic programming approach, we show that for any fixed ℓ, MWIS can be solved for ℓclaw-free graphs in polynomial time. This solves the open case for MWIS on (P_3+claw)-free graphs and ($2P_2$+claw)-free graphs, and extends known results for MWIS on claw-free graphs, ℓK_2-free graphs for any fixed ℓ, (K_2+claw)-free graphs, ($2P_2$+claw)-free graphs, $2P_3$-free graphs and more generally, for ℓP_3-free graphs.

1.2 Maximal Independent Sets in $2K_2$-Free Graphs

Our approach for MWIS on ℓclaw-free graphs is based on the following Algorithm Alpha (called Algorithm $\mathcal{A}$ in [23]) which formulates Farber's argument [12] for showing that $2K_2$-free graphs have $\mathcal{O}(n^2)$ maximal independent sets.

For a graph G, with vertex set $\{v_1, v_2, \ldots, v_n\}$, let us write $G_i := G[\{v_1, v_2, \ldots, v_i\}]$. At each iteration i, $1 \leq i \leq n$, Algorithm Alpha provides a family $\mathcal{S}_i$ of subsets of $\{v_1, v_2, \ldots, v_i\}$ 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 $\{v_1, v_2, \ldots, v_n\}$.
Output: A family $\mathcal{S}$ of independent sets of G that contains all maximal independent sets of G and is such that $|\mathcal{S}| = \mathcal{O}(n^2)$.

$\mathcal{S} := \{\emptyset\}$;
For $i =: 1$ **to** n **do**
begin
1. [Extension of some members of $\mathcal{S}$]

For each $H \in \mathcal{S}$ **do**
If $H \cup \{v_i\}$ is an independent set **then** $H := H \cup \{v_i\}$.
2. [Addition of new members to $\mathcal{S}$]
For each K_2 of G_i containing v_i, i.e., for each edge uv_i of G_i **do**
$\mathcal{S} := \mathcal{S} \cup \{\{v_i\} \cup A_{G_i}(u, v_i)\}$.
end.

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 [12] *For $2K_2$-free graphs, the MWIS problem can be solved in time $\mathcal{O}(n^4)$.*

2 A Basic Lemma

Now we provide a basic result which leads to a polynomial number of subcases to obtain all maximal independent sets. For each $k \in \{1, \ldots, 14\}$, let L_k be the graph drawn in Figure 1. Note that each L_k contains an induced claw. For each $k \in \{1, \ldots, 14\}$, let $V(L_k)$ denotes the vertex set of L_k, let $W(L_k)$ denotes the set of white vertices of L_k, let $B(L_k)$ denotes the set of black vertices of L_k, and let *top*(L_k) denotes the (white) vertex at the top of L_k.

Lemma 1 *For a graph $G = (V, E)$, assume that the vertex $v \in V$ is contained in an induced claw of G and $G[V \setminus \{v\}]$ is claw-free. Then for each maximal independent set I of G with $v \in I$, there is a $k \in \{1, \ldots, 14\}$ such that $I \subseteq W(L_k) \cup A_G(L_k)$ for an induced subgraph L_k of G with $v = top(L_k)$.*

Proof. Let K be a claw in G containing v, say, with vertex set $\{v, a, b, c\}$. Let I be a maximal independent set of G containing v, and let $I' := I \setminus \{v\}$. Then for $U := V \setminus \{v\}$, I' is a maximal independent set of $G[U \setminus N(v)]$. Let us distinguish between the following cases.

Case 1 v **is the center of** K.

Since $G[U]$ is claw-free, each of a, b, c has at most two neighbors in I'.

Case 1.1 If a vertex of a, b, c, say a, has two neighbors in I', say s_1, s_2 then $I \subseteq W(L_1) \cup A_G(L_1)$ with $V(L_1) = \{v, a, s_1, s_2\}$, $W(L_1) = \{v, s_1, s_2\}$, and $v = \text{top}(L_1)$.

Case 1.2 If none of a, b, c has a neighbor in I' then $I \subseteq W(L_2) \cup A_G(L_2)$ with $V(L_2) = \{v, a, b, c\}$ and $v = \text{top}(L_2)$.

Case 1.3 Now assume that Cases 1.1 and 1.2 are excluded. This means that one of a, b, c, say without loss of generality a, has exactly one neighbor in I' and b and c have at most one neighbor in I'. Let $as_1 \in E$ for $s_1 \in I'$. Note that at most one of b and

c is adjacent to s_1 since $G[U]$ is claw-free, and no vertex of I' is adjacent to all of a, b and c.

If $N(b) \cap I' = N(c) \cap I' = \emptyset$ then we have $I \subseteq W(L_3) \cup A_G(L_3)$ with $V(L_3) = \{v, a, b, c, s_1\}$ and $v = \text{top}(L_3)$.

If b has exactly one neighbor in I', say s_2, and $N(c) \cap I' = \emptyset$ then if $s_1 \neq s_2$, we have $I \subseteq W(L_4) \cup A_G(L_4)$ with $V(L_4) = \{v, a, b, c, s_1, s_2\}$ and $v = \text{top}(L_4)$, and if $s_1 = s_2$, we have $I \subseteq W(L_6) \cup A_G(L_6)$ with $V(L_6) = \{v, a, b, c, s_1\}$ and $v = \text{top}(L_6)$, and similarly for the case when c has exactly one neighbor in I', and $N(b) \cap I' = \emptyset$.

Finally, assume that each of b and c has a neighbor in I', i.e., there are $s_2, s_3 \in I'$ with $bs_2 \in E$ and $cs_3 \in E$.

If s_1, s_2, s_3 are pairwise distinct then we have $I \subseteq W(L_5) \cup A_G(L_5)$ with $V(L_5) = \{v, a, b, c, s_1, s_2, s_3\}$ and $v = \text{top}(L_5)$.

Now assume that $|\{s_1, s_2, s_3\}| = 2$ (recall that $|\{s_1, s_2, s_3\}| = 1$ is impossible). Without loss of generality, let $s_1 = s_2$. Then we have $I \subseteq W(L_7) \cup A_G(L_7)$ with $V(L_7) = \{v, a, b, c, s_1, s_3\}$ and $v = \text{top}(L_7)$.

Case 2 v **is a leaf of** K.

Without loss of generality, let b be the center of K. Since $G[U]$ is claw-free, b has at most two neighbors in I', and if $a \notin I'$ ($c \notin I'$, respectively), the same holds for a (c, respectively). The following subcases are exhaustive by symmetry.

Case 2.1 If $a, c \in I'$ then $I \subseteq W(L_1) \cup A_G(L_1)$ with $V(L_1) = \{v, a, b, c\}$ and $v = \text{top}(L_1)$.

Case 2.2 If exactly one of a, c is in I', say without loss of generality, $a \in I'$ and $c \notin I'$ (and more generally, only one of the neighbors of b is in I' - otherwise we have Case 2.1) then c has a neighbor in I', say s, since I' is a maximal independent set of $G[U \setminus N(v)]$. Then clearly, s is nonadjacent to a, and a is the unique neighbor of b in I' (otherwise b would have two neighbors in I'), and hence (since $G[U]$ is claw-free) s is the unique neighbor of c in I'. Then $I \subseteq W(L_8) \cup A_G(L_8)$ with $V(L_8) = \{v, a, b, c, s\}$ and $v = \text{top}(L_8)$.

Case 2.3 Now assume that Cases 2.1 and 2.2 are excluded. Thus, $a, c \notin I'$. Then each of a and c must have a neighbor in I' since I' is a maximal independent set of $G[U \setminus N(v)]$.

First assume that no neighbor of a or c in I' is adjacent to b. Then each of a and c has exactly one neighbor in I' and b has no neighbor in I', else a claw in $G[U]$ would arise involving b. Let $s_1, s_2 \in I'$ with $as_1 \in E$, $cs_2 \in E$.

If $s_1 \neq s_2$ then $I \subseteq W(L_9) \cup A_G(L_9)$ with $V(L_9) = \{v, a, b, c, s_1, s_2\}$ and $v = \text{top}(L_9)$.

If $s_1 = s_2$ then $I \subseteq W(L_{10}) \cup A_G(L_{10})$ with $V(L_{10}) = \{v, a, b, c, s_1\}$ and $v = \text{top}(L_{10})$.

Now assume that, without loss of generality, a neighbor $s \in I'$ of a is adjacent to b. We claim:

(*i*) s is adjacent to c, since otherwise Case 2.2 holds with s instead of a;

(*ii*) s is the unique neighbor of b in I' (since otherwise Case 2.1 holds);

(*iii*) a and c each have at most one more neighbor in I' (since $G[U]$ is claw-free).

If neither a nor c have another neighbor in I' then $I \subseteq W(L_{11}) \cup A_G(L_{11})$ with $V(L_{11}) = \{v, a, b, c, s\}$ and $v = \text{top}(L_{11})$.

If there is $s_1 \in I'$ with $s_1 \neq s$, $as_1 \in E$ and the only neighbor of c in I' is s then $I \subseteq W(L_{12}) \cup A_G(L_{12})$ with $V(L_{12}) = \{v, a, b, c, s, s_1\}$ and $v = \text{top}(L_{12})$, and similarly if c has two neighbors $s, s_1 \in I'$ and a has only neighbor $s \in I'$.

Finally, if a and c both have another neighbor in I', say $s_1, s_2 \in I'$, $s \neq s_1, s \neq s_2$ with $as_1 \in E$ and $cs_2 \in E$ then we have:

If $s_1 \neq s_2$ then $I \subseteq W(L_{13}) \cup A_G(L_{13})$ with $V(L_{13}) = \{v, a, b, c, s, s_1, s_2\}$ and $v = \text{top}(L_{13})$, and if $s_1 = s_2$ then $I \subseteq W(L_{14}) \cup A_G(L_{14})$ with $V(L_{14}) = \{v, a, b, c, s, s_1\}$ and $v = \text{top}(L_{14})$. □

3 MWIS for (Claw+Claw)-Free Graphs

Now we show that for (claw+claw)-free graphs, MWIS can be solved in time $\mathcal{O}(n^{10})$. For this, we need the following notion:

Definition 1 *Let $G = (V, E)$ be a graph and let $\mathcal{S}$ be a family of subsets of V. Then $\mathcal{S}$ is a* good claw-free family of G *if the following holds:*

(*i*) *Each member of $\mathcal{S}$ induces a claw-free subgraph in G.*

(*ii*) *Each maximal independent set of G is contained in some member of $\mathcal{S}$.*

The following Algorithm Gamma(2) computes a good claw-free family $\mathcal{S}$ of any input (claw+claw)-free graph G. Recall that $G_i := G[\{v_1, v_2, \dots, v_i\}]$ for a graph G with vertex set $\{v_1, v_2, \dots, v_n\}$.

Algorithm Gamma(2)
Input: A (claw+claw)-free graph G with vertex set $\{v_1, v_2, \dots, v_n\}$.
Output: A good claw-free family $\mathcal{S}$ of G such that $|\mathcal{S}| = \mathcal{O}(n^7)$.

$\mathcal{S} := \{\emptyset\}$;
For $i =: 1$ **to** n **do**
begin
1. [Extension of some members of $\mathcal{S}$]
 For each $U \in \mathcal{S}$ **do**
 If $G[U \cup \{v_i\}]$ is claw-free **then** $U := U \cup \{v_i\}$.
2. [Addition of new members to $\mathcal{S}$]

For each induced L_k of G_i, $k \in \{1, \dots, 14\}$, with $v_i = \text{top}(L_k)$ **do**
$\mathcal{S} := \mathcal{S} \cup \{W(L_k) \cup A_{G_i}(L_k)\}$.
end.

Theorem 4 *Algorithm Gamma(2) correctly produces the desired output, and can be implemented to run in $\mathcal{O}(n^{10})$ time.*

Proof. First let us observe that for every $i \in \{1, \dots, n\}$ and $k \in \{1, \dots, 14\}$, $G[A_{G_i}(L_k)]$ is claw-free since $G[L_k]$ contains an induced claw and G is (claw+claw)-free.

Then let us prove the following claims.

Claim 1 *The following statements hold:*

(*i*) *Each member of $\mathcal{S}$ induces a claw-free subgraph of G.*

(*ii*) *Each maximal independent set of G is contained in some member of $\mathcal{S}$.*

Proof. (*i*): Each member of $\mathcal{S}$ is created either in the initialization step as the empty set, or in Step 1 or in Step 2 of some iteration. Each member of $\mathcal{S}$ created in Step 1 is an extension of a member of $\mathcal{S}$ and induces a claw-free subgraph of G by definition of the step. Each member of $\mathcal{S}$ created in Step 2 is the disjoint union of $W(L_k)$ and $A_{G_i}(L_k)$, both inducing a claw-free subgraph of G, and then induces a claw-free subgraph of G. This completes the proof of statement (*i*).

(*ii*): By $\mathcal{S}_i$, let us denote the family $\mathcal{S}$ resulting by the i-th iteration of Algorithm Gamma(2). Let us show that for all $i \in \{1, \dots, n\}$, each maximal independent set of G_i is contained in a member H of $\mathcal{S}_i$. The proof is done by induction. For $i = 1$, the statement is trivial. Then let us assume that the statement holds for $i - 1$ and prove that it holds for i.

Let I be a maximal independent set of G_i.

If $v_i \notin I$, then by the induction assumption, 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 a (not necessarily proper) subset of a member of $\mathcal{S}_i$.

If $v_i \in I$, then let us consider the following argument. By the induction assumption, let $U \in \mathcal{S}_{i-1}$ with $I \setminus \{v_i\} \subseteq U$. Note that for all j, $1 \leq j \leq n$, each member of $\mathcal{S}_j$ induces a claw-free graph, as one can easily verify by an argument similar to the proof of statement (*i*). Thus, $G[U]$ is claw-free.

Then let us consider the following two cases which are exhaustive by definition of Algorithm Gamma(2).

Case 1 $G[U \cup \{v_i\}]$ is claw-free.

Then I is contained in the set $U \cup \{v_i\}$, which is a member of $\mathcal{S}_i$ since it is generated by Step 1 of the algorithm at iteration i.

Case 2 $G[U \cup \{v_i\}]$ is not claw-free.

Then by Lemma 1, since $G[U]$ is claw-free, there is a $k \in \{1, \ldots, 14\}$ such that $I \subseteq W(L_k) \cup A_{G[U \cup \{v_i\}]}(L_k) \subseteq W(L_k) \cup A_{G_i}(L_k)$ for an induced subgraph L_k of G_i with $v_i = \text{top}(L_k)$, and $W(L_k) \cup A_{G_i}(L_k)$ is contained in $\mathcal{S}_i$ since it is generated by Step 2 of Algorithm Gamma(2) at iteration i. This shows statement (ii) of Claim 1. □

Claim 2 *$|\mathcal{S}|$ is of $\mathcal{O}(n^7)$ and $\mathcal{S}$ can be computed in $\mathcal{O}(n^{10})$ time.*

Proof. $|\mathcal{S}|$ is clearly bounded by the number of induced L_k's in G, so it is in $\mathcal{O}(n^7)$. Then $\mathcal{S}$ can be computed in $\mathcal{O}(n^{10})$ time: in fact, referring to Step 1, note that to check if $G[U \cup \{v_i\}]$ is claw-free ($G[U]$ being claw-free) can be done in $\mathcal{O}(n^3)$ time; referring to Step 2, note that the algorithm produces the anti-neighborhoods of all induced L_k's of G_i just once since at iteration i all such L_k's have $\text{top}(L_k) = v_i$, and that computing $A_{G_i}(L_k)$ can be done in $\mathcal{O}(n)$ time. □

This completes the proof of Theorem 4. □

Theorem 5 *For (claw+claw)-free graphs, the MWIS problem can be solved in $\mathcal{O}(n^{10})$ time.*

Proof. For every (claw+claw)-free graph G, the MWIS problem can be solved by the following algorithm:

(1) Execute Algorithm Gamma(2) for G with the resulting good claw-free family $\mathcal{S}$ of G.

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

Correctness: By Theorem 4, the algorithm is correct.

Time bound: By Theorem 4, $|\mathcal{S}| = \mathcal{O}(n^7)$ and $\mathcal{S}$ can be computed in $\mathcal{O}(n^{10})$ time. Then step (1) can be executed in $\mathcal{O}(n^9)$ time. Then, by Theorem 1, step (2) can be executed in $\mathcal{O}(n^9 \log(n))$ time. Thus, the algorithm can be executed in $\mathcal{O}(n^{10})$ time. □

4 MWIS for ℓClaw-Free Graphs

In this section we show that for any fixed $\ell \geq 2$, MWIS for ℓclaw-free graphs can be solved in polynomial time. For this, we first describe the subsequent Algorithm Gamma(ℓ), which for any ℓclaw-free input graph G computes a good claw-free family $\mathcal{S}$ of G. The approach recursively uses Algorithm Gamma($\ell-1$) for Algorithm Gamma(ℓ), starting with Algorithm Gamma(2) of section 3.

Algorithm Gamma(ℓ)
Input: An ℓclaw-free graph G with vertex set $\{v_1, v_2, \dots, v_n\}$.
Output: A good claw-free family $\mathcal{S}$ of G such that $\mathcal{S}$ has polynomially many members.

$\mathcal{S} := \{\emptyset\}$;
For $i =: 1$ **to** n **do**
begin
1. [Extension of some members of $\mathcal{S}$]
   For each $U \in \mathcal{S}$ **do**
   If $G[U \cup \{v_i\}]$ is claw-free **then** $U := U \cup \{v_i\}$.
2. [Addition of new members to $\mathcal{S}$]
   For each induced L_k of G_i, $k \in \{1, \dots, 14\}$, with $v_i = \text{top}(L_k)$ **do**
      Execute Algorithm Gamma($\ell - 1$) for $G[A_{G_i}(L_k)]$
      with the resulting good claw-free family, say $\mathcal{F}$, of $G[A_{G_i}(L_k)]$.
      For each $F \in \mathcal{F}$, set $\mathcal{S} := \mathcal{S} \cup \{W(L_k) \cup F\}$.

end.

Remark 1 *For $\ell = 2$, Algorithm Gamma(ℓ) is exactly Algorithm Gamma(2) of section 3. In fact, for $\ell = 2$, $G[A_{G_i}(L_k)]$ is claw-free, i.e., a good claw-free family of $G[A_{G_i}(L_k)]$ is trivially formed by one member, namely, $A_{G_i}(L_k)$.*

Theorem 6 *For any fixed $\ell \geq 2$, Algorithm Gamma(ℓ) correctly produces the desired output, and can be implemented to run in polynomial time.*

Proof. For $\ell = 2$, Theorem 6 holds, since by Remark 1, it corresponds exactly to Theorem 4. For $\ell > 2$ we will prove Theorem 6 by induction, that is, we assume that it is true for $\ell - 1$ and we will show that it holds for ℓ.

First let us observe that for all $i \in \{1, \dots, n\}$ and $k \in \{1, \dots, 14\}$, $G[A_{G_i}(L_k)]$ is $(\ell - 1)$claw-free, since $G[L_k]$ contains an induced claw and G is ℓclaw-free.

Then let us prove the following claims.

Claim 3 *The following statements hold:*

(i) Each member of $\mathcal{S}$ induces a claw-free subgraph of G.

(ii) Each maximal independent set of G is contained in some member of $\mathcal{S}$.

Proof. (i): The proof is similar to that of Claim 1(i) in the proof of Theorem 4, by definition of good claw-free family and by the induction assumption on ℓ.

(ii): By $\mathcal{S}_i$, let us denote the family $\mathcal{S}$ resulting by the i-th iteration of Algorithm Gamma(ℓ). Let us show that for all $i \in \{1, \dots, n\}$, each maximal independent set of G_i is contained in a member H of $\mathcal{S}_i$. The proof is done by induction. For $i = 1$, the statement is trivial. Then let us assume that the statement holds for $i-1$ and prove that it holds for i.

Let I be a maximal independent set of G_i.

If $v_i \notin I$, then by the induction assumption, 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 a (not necessarily proper) subset of a member of $\mathcal{S}_i$.

If $v_i \in I$, then let us consider the following argument. By the induction assumption, let $U \in \mathcal{S}_{i-1}$ with $I \setminus \{v_i\} \subseteq U$. Note that for all j, $1 \leq j \leq n$, each member of $\mathcal{S}_j$ induces a claw-free graph, as one can easily verify by an argument similar to the proof of statement (i). Thus, $G[U]$ is claw-free.

Then let us consider the following two cases which are exhaustive by definition of Algorithm Gamma(ℓ).

Case 1 $G[U \cup \{v_i\}]$ is claw-free.

Then I is contained in the set $U \cup \{v_i\}$, which is a member of $\mathcal{S}_i$ since it is generated by Step 1 of the algorithm at iteration i.

Case 2 $G[U \cup \{v_i\}]$ is not claw-free.

Then by Lemma 1, since $G[U]$ is claw-free, there is a $k \in \{1, \ldots, 14\}$ such that $I \subseteq W(L_k) \cup A_{G[U \cup \{v_i\}]}(L_k) \subseteq W(L_k) \cup A_{G_i}(L_k)$ for an induced subgraph L_k of G_i with $v_i = \text{top}(L_k)$; on the other hand each maximal independent set of $G[W(L_k) \cup A_{G_i}(L_k)]$ is contained in some member of $\mathcal{S}_i$ by Step 2 of Algorithm Gamma(ℓ) at iteration i, by definition of good claw-free family and by the induction assumption on ℓ; then I is contained in some member of $\mathcal{S}_i$. This shows statement (ii) of Claim 3. □

Claim 4 *The family $\mathcal{S}$ contains polynomially many members and can be computed in polynomial time.*

Proof. The family $\mathcal{S}$ contains polynomially many members, since the number of induced L_k's in G is of $\mathcal{O}(n^7)$, and since by the above inductive assumption the good claw-free family of $G[A_{G_i}(L_k)]$ provided by Algorithm Gamma($\ell - 1$) contains polynomially many members. Then $\mathcal{S}$ can be computed in polynomial time: in fact, referring to Step 1, note that to check if $G[U \cup \{v_i\}]$ is claw-free ($G[U]$ being claw-free) can be done in $\mathcal{O}(n^3)$ time; referring to Step 2, note that the algorithm produces the anti-neighborhoods of all L_k's of G_i just once since at iteration i all such L_k's have $\text{top}(L_k) = v_i$, and that to compute $A_{G_i}(L_k)$ can be done in $\mathcal{O}(n)$ time. □

This completes the proof of Theorem 6. □

Theorem 7 *For any fixed ℓ, for ℓclaw-free graphs, the MWIS problem can be solved in polynomial time.*

Proof. For any fixed ℓ the MWIS problem can be solved for every ℓclaw-free graph G by the following algorithm:

(1) Execute Algorithm Gamma(ℓ) for G with the resulting good claw-free family $\mathcal{S}$ of G.

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

Correctness: By Theorem 6, the algorithm is correct.

Time bound: By Theorem 6, the family $\mathcal{S}$ contains polynomially many members and can be computed in polynomial time. Then step (1) can be executed in polynomial time. Then, by Theorem 1, step (2) can be executed in polynomial time. Thus, the algorithm can be executed in polynomial time. □

5 Conclusion

The main result of this paper is Theorem 7 showing that for any fixed ℓ, for ℓclaw-free graphs, the MWIS problem can be solved in polynomial time by a dynamic programming approach. In particular as a corollary of the above one obtains:

Corollary 1 *For any fixed ℓ there is a polynomial-time algorithm which, for every ℓclaw-free graph G, computes a family $\mathcal{S}$ of subsets of $V(G)$ inducing claw-free graphs (with $\mathcal{S}$ containing polynomially many members) such that every maximal independent set of G is contained in some member of $\mathcal{S}$.*

Let us conclude with three comments.

1. Corollary 1 still holds by replacing the claw with any induced subgraph H of the claw, that is apart from trivial induced subgraphs, with $H = K_2$ (obtaining in this way a result similar to that for ℓK_2-free graphs [2, 5, 13, 31]) or with $H = P_3$. In particular Lemma 1 can be reformulated by replacing the claw with H, and by replacing the family of graphs L_k's with one graph isomorphic to K_2 (for $H = K_2$) or with two graphs isomorphic to P_3 (for $H = P_3$).

2. It would be interesting to apply the dynamic programming approach for more general cases as well, say for solving more open problems and extending previous polynomial results, but so far we do not see how to apply it e.g. for ℓfork-free graphs or ℓapple-free graphs.

3. Theorem 7 and the known results imply that the new graph classes, defined by forbidding one induced subgraphs, for which the complexity MWIS problem seems to be open are: P_7-free graphs, $S_{1,1,3}$-free graphs, $S_{1,2,2}$-free graphs, $K_2 + P_4$-free graphs.

References

[1] V.E. Alekseev, On the local restriction effect on the complexity of finding the graph independence number, *Combinatorial-algebraic Methods in Applied Mathematics*, Gorkiy University Press, Gorkiy (1983) 3-13 (in Russian).

[2] V.E. Alekseev, On the number of maximal independent sets in graphs from hereditary classes, *Combinatorial-algebraic Methods in Discrete Optimization*, Gorkiy University Press, Gorkiy (1991) 5-8 (in Russian).

[3] V.E. Alekseev, A polynomial algorithm for finding largest independent sets in fork-free graphs, *Discrete Analysis and Operations Research* Ser. 1, 6 (1999) 3-19 (in Russian), *Discrete Applied Mathematics* 135 (2004) 3-16.

[4] V.E. Alekseev, On easy and hard hereditary classes of graphs with respect to the independent set problem, *Discrete Applied Mathematics* 132 (2004) 17-26.

[5] E. Balas and C.S. Yu, On graphs with polynomially solvable maximum-weight clique problem, *Networks* 19 (1989) 247-253.

[6] A. Brandstädt, V.B. Le, and J.P. Spinrad, Graph Classes: A Survey, *SIAM Monographs on Discrete Math. Appl., Vol.* 3, SIAM, Philadelphia (1999).

[7] A. Brandstädt, V.V. Lozin, and R. Mosca, Independent Sets of Maximum Weight in Apple-Free Graphs, *SIAM J. Discrete Math.* 24 (2010) 239-254.

[8] A. Brandstädt, T. Klembt, V.V. Lozin, and R. Mosca, Independent Sets of Maximum Weight in Apple-Free Graphs, ISAAC 2008, LNCS 5369 (2008).

[9] D.G. Corneil, Y. Perl, and L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Computing* 14 (1985) 926-934.

[10] Y. Faenza, G. Oriolo, and G. Stauffer, An algorithmic decomposition of claw-free graphs leading to an O(n^3)-algorithm for the weighted independent set problem, Extended abstract in: Proceedings SODA 2011, pp. 630-646.

[11] Y. Faenza, G. Oriolo, and G. Stauffer, Solving the Weighted Stable Set Problem in Claw-Free Graphs via Decomposition. *J. ACM* 61 (4) Article No. 20, pp. 1-41, 2014.

[12] M. Farber, On diameters and radii of bridged graphs, *Discrete Mathematics* 73 (1989) 249-260.

[13] M. Farber, M. Hujter, and Zs. Tuza, An upper bound on the number of cliques in a graph, *Networks* 23 (1993) 75-83.

[14] G.H. Fricke, S.T. Hedetniemi, and D.P. Jacobs, Independence and irredundance in k-regular graphs, *Ars Combinatoria* 49 (1998) 271-279.

[15] M.R. Garey, D.S. Johnson, and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoretical Computer Science* 1 (1976) 237-267.

[16] M.R. Garey and D.S. Johnson, The rectilinear Steiner tree problem is NP-complete, *SIAM J. Applied Mathematics* 32 (1977) 826-834.

[17] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completness, Freeman, San Francisco, CA (1979).

[18] M. Grötschel, L. Lovász, and A. Schrijver, Polynomial algorithms for perfect graphs, *Annals of Discrete Mathematics* 21 (1984) 325-356.

[19] Andrzej Grzesik, Tereza Klimoşová, Marcin Pilipczuk, Micha Pilipczuk, Polynomial-time algorithm for Maximum Weight Independent Set on P_6-free graphs, arXiv:1707.05491

[20] D. Lokshtanov, M. Vatshelle, and Y. Villanger, Independent Sets in P_5-free Graphs in Polynomial Time, http://www.ii.uib.no/∼martinv/Papers/ISinP5free.pdf; extended abstract in: Proceedings of SODA 2014: 570-581.

[21] V.V. Lozin, From matchings to independent sets, *Discrete Applied Mathematics* 231 (2017) 4-14.

[22] V.V. Lozin and M. Milanič, A polynomial algorithm to find an independent set of maximum weight in a fork-free graph, *J. Discrete Algorithms* 6 (2008) 595-604..

[23] V.V. Lozin and R. Mosca, Independent sets in extensions of $2K_2$-free graphs, *Discrete Applied Mathematics* 146 (2005) 74-80.

[24] V.V. Lozin and R. Mosca, Maximum regular subgraphs in $2P_3$-free graphs, *Theoretical Computer Science* 460 (2012) 26-33.

[25] G.J. Minty, On maximal independent sets of vertices in claw-free graphs, *J. Combinatorial Theory, Series B* 28 (1980) 284-304.

[26] O.J. Murphy, Computing independent sets in graphs with large girth, *Discrete Applied Mathematics* 35 (1992) 167-170.

[27] D. Nakamura and A. Tamura, A revision of Minty's algorithm for finding a maximum weight stable set in a claw-free graph, *J. Operations Research Society of Japan* 44 (2001) 194-204.

[28] P. Nobili and A. Sassano, A reduction algorithm for the weighted stable set problem in claw-free graphs, Extended abstract in: Proceedings CTW 2011: 223-226.

[29] P. Nobili and A. Sassano, An $\mathcal{O}(n^2 \log n)$ algorithm for the weighted stable set problem in claw-free graphs. CoRR abs/1501.05775 (2015).

[30] S. Poljak, A note on stable sets and colorings of graphs, *Commun. Math. Univ. Carolinae* 15 (1974) 307-309.

[31] E. Prisner, Graphs with few cliques, Graph Theory, Combinatorics and Algorithms, vol. 1, 2 (Kalamazoo, MI, 1992), Wiley-Interscience Publishers, Wiley, New York, 1995, 945-956.

[32] N. Sbihi, Algorithme de recherche d'un independent de cardinalité maximum dans un graphe sans étoile, *Discrete Mathematics* 29 (1980) 53-76.

[33] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, A new algorithm for generating all maximal independent sets, *SIAM J. on Computing* 6 (1977) 505-517.