HAL open science

On the complexity of the exact weighted independent set problem

Martin Milanic, Jérôme Monnot

To cite this version:

Martin Milanic, Jérôme Monnot. On the complexity of the exact weighted independent set problem. 2007. hal-00178135

HAL Id: hal-00178135

https://hal.science/hal-00178135v1

Preprint submitted on 10 Oct 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE PARIS DAUPHINE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision

CNRS UMR 7024

CAHIER DU LAMSADE

258

Juillet 2007

On the complexity of the exact weighted independent set problem

Martin Milanič, Jérôme Monnot

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

On the complexity of the exact weighted independent set problem

Martin Milanič[*] *Jérôme Monnot*[†‡]

Abstract

In this paper, we introduce the *exact weighted independent set* problem (EWIS), the problem of determining whether a given weighted graph contains an independent set of a given weight. Our motivation comes from the related exact perfect matching problem, whose computational complexity is still unknown. We determine the complexities of the EWIS problem and its restricted version EWIS_α (where the independent set is required to be of maximum size) for several graph classes. These problems are strongly **NP**-complete for cubic bipartite graphs; we also extend this result to a more general setting. On the positive side, we show that EWIS and EWIS_α can be solved in pseudo-polynomial time for chordal graphs, AT-free graphs, distance-hereditary graphs, circle graphs, graphs of bounded clique-width, and several subclasses of P_5-free and fork-free graphs. In particular, we show how modular decomposition can be applied to the exact weighted independent set problem.

Keywords: exact weighted independent set, **NP**-complete, pseudo-polynomial algorithm, modular decomposition.

1 Introduction

Suppose we have a well-solved optimization problem, such as minimum spanning tree, maximum cut in planar graphs, minimum weight perfect matching, or maximum weight independent set in a bipartite graph. How hard is it to determine whether there exists a solution with a given weight? Papadimitriou and Yannakakis showed in [42] that these so-called *exact* versions of the above optimization problems are **NP**-complete when the weights are encoded in binary. The question is then, what happens if the weights are "small," i.e., encoded in unary? Contrary to the binary case, the answer to this question depends on the problem.

- The *exact spanning tree* problem, and more generally, the *exact arborescence* problem are solvable in pseudo-polynomial time [5].
- The *exact cut* problem is solvable in pseudo-polynomial time for planar and toroidal graphs [5].
- The *exact perfect matching* problem is solvable in pseudo-polynomial time for planar graphs [5], and more generally, for graphs that have a Pfaffian orientation (provided

[*]RUTCOR, Rutgers University, 640 Bartholomew Rd, Piscataway NJ, 08854-8003, USA. E-mail: `mmilanic@rutcor.rutgers.edu`

[†]LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, F-75775 Paris Cedex 16, France. E-mail: `monnot@lamsade.dauphine.fr`

[‡]Part of this research was carried out while the first author was visiting LAMSADE. Financial support of DIMACS, as well as hospitality of members of LAMSADE are gratefully acknowledged.

one is given).[1] Karzanov [30] gives a polynomial-time algorithm for the special case of the exact perfect matching problem, when the graph is either complete or complete bipartite, and the weights are restricted to 0 and 1. Papadimitriou and Yannakakis show in [42] that the problem for general (or bipartite) graphs with weights encoded in unary is polynomially reducible to the one with 0-1 weights. Mulmuley, Vazirani and Vazirani [38] show that the exact perfect matching problem has a randomized pseudo-polynomial-time algorithm. However, the deterministic complexity of this problem remains unsettled, even for bipartite graphs. (Papadimitriou and Yannakakis conjectured that it is **NP**-complete [42].)

The exact perfect matching problem is of great practical importance. It has applications in such diverse areas as bus-driver scheduling, statistical mechanics (see [31]), DNA sequencing [7], and robust assignment problems [21]. The problem consists of determining whether a given edge weighted graph contains a perfect matching of a given weight. The fact that its complexity is still unknown motivates us to introduce and study the *exact weighted independent set* problem and a restricted version of it, both closely related to the exact perfect matching problem.

An *independent set* (sometimes called *stable set*) in a graph is a set of pairwise non-adjacent vertices. The *weighted independent set* problem (WIS) asks for an independent set of maximum weight in a given weighted graph (G, w). If all weights are the same, we speak about the *independent set* problem (IS), which consists in finding an independent set of maximum cardinality. The optimal values of these problems are denoted by $\alpha_w(G)$ and $\alpha(G)$, respectively.

The *exact weighted independent set* problem (EWIS) consists of determining whether a given weighted graph (G, w) with $G = (V, E)$ and $w : V \to \mathbb{Z}$ contains an independent set whose total weight (i.e., the sum of the weights of its members) equals a given integer M. The *exact weighted maximum independent set* problem (EWIS_α) is the restriction of EWIS where we require the independent set to be a maximum independent set of the graph. Thus, given a weighted graph (G, w) and an integer M, EWIS_α asks about the existence of an independent set I of G with $|I| = \alpha(G)$ and $w(I) = M$.

The connection between the exact perfect matching problem and the exact weighted independent set problem is best understood through line graphs. The *line graph* $L(G)$ of a graph $G = (V, E)$ is the graph whose vertex set is E, and whose two vertices are adjacent if and only if they share a common vertex as edges of G. Clearly, there is a one-to-one correspondence between the matchings of a graph and the independent sets of its line graph. Thus, the exact matching problem is precisely the exact weighted independent set problem, restricted to the class of line graphs. Similarly, under the (polynomially verifiable) assumption that the input graph has a perfect matching, the exact perfect matching problem is precisely the exact weighted maximum independent set problem, restricted to the class of the line graphs of graphs with a perfect matching.

Our contributions. We focus on the problem of determining the complexities of EWIS and EWIS_α for particular graph classes. On one hand, we present the first nontrivial strong **NP**-completeness result for these problems. On the other hand, we distinguish several classes of graphs they can be solved in pseudo-polynomial time.

The main results of this paper can be summarized in the spirit of the above list of complexity results on exact problems:

- The *exact weighted independent set* and the *exact weighted maximum independent set*

[1] Bipartite graphs with a Pfaffian orientation have been characterized in [49], where a polynomial-time recognition algorithm is also presented.

problems are strongly **NP**-complete for cubic bipartite graphs.[2]

- The *exact weighted independent set* and the *exact weighted maximum independent set* problems are solvable in pseudo-polynomial time for any of the following graph classes:
  - *mK_2-free graphs*,
  - *interval graphs* and their generalizations *k-thin graphs*,
  - *chordal graphs*,
  - *AT-free graphs*, and
  - *(claw, net)-free graphs*,
  - *distance-hereditary graphs*,
  - *circle graphs*,
  - *graphs of bounded treewidth*,
  - *graphs of bounded clique-width*,
  - certain *subclasses of P_5-free and fork-free graphs*.

Of independent interest, we show how modular decomposition can be applied to the exact weighted independent set problem.

Note that in view of the relation between the exact perfect matching problem and the exact weighted maximum independent set problem, each of the above polynomial results also gives a polynomial result for the exact perfect matching problem. Whenever EWIS_α is (pseudo-)polynomially solvable for a class of graphs $\mathcal{G}$, the exact perfect matching problem is (pseudo-)polynomially solvable for graphs in the set $\{G : L(G) \in \mathcal{G}\}$. For example,

- The *exact perfect matching* problem is solvable in pseudo-polynomial time for graphs of bounded treewidth.

Notation and organization. All graphs considered are finite, simple and undirected. Unless otherwise stated, n and m will denote the number of vertices and edges, respectively, of the graph considered. As usual, P_n and C_n denote the chordless path and the chordless cycle on n vertices. For a graph G, we will denote by $V(G)$ and $E(G)$ the vertex-set and the edge-set of G, respectively. Individual edges will be denoted by square brackets: an edge with endpoints u and v will be denoted by $[u, v]$. For a vertex x in a graph G, we denote by $N_G(x)$ the neighborhood of x in G, i.e., the set of vertices adjacent to x, and by $N_G[x]$ the closed neighborhood of x, i.e., the set $N_G(x) \cup \{x\}$. We will write $N(x)$ and $N[x]$ instead of $N_G(x)$ and $N_G[x]$ if no confusion can arise. For a graph G, we denote by co-G (also $\overline{G}$) the edge-complement of G. By K_n we denote the complete graph on n vertices, and by $K_{s,t}$ the complete bipartite graph with parts of size s and t. By component we will always mean a connected component. For graph-theoretical terms not defined here, the reader is referred to Berge's book [6].

The triple (G, w, M) will always represent an instance of EWIS (or EWIS_α), i.e., $G = (V, E)$ is a graph, $w : V \to \mathbb{Z}$ are vertex weights, and $M \in \mathbb{Z}$ is the target weight. If H is an induced subgraph of G, we will also consider triples of the form (H, w, M) as instances of EWIS, with the weights w representing the restriction of w to $V(H)$. We will denote by $\text{EWIS}(G, w, M)$ the solution to the instance (G, w, M) of EWIS, that is, $\text{EWIS}(G, w, M)$ is *yes* if there is an independent set I in G with $w(I) = M$, and *no* otherwise. Similarly, $\text{EWIS}_\alpha(G, w, M)$ is *yes* if there is a maximum independent set I in G with $w(I) = M$, and *no*

[2] We also strengthen this result considerably, however we postpone the detailed formulation until Section 3.2.

otherwise. Finally, for a subset of vertices $V' \subseteq V$, we let $w(V') = \sum_{v \in V'} w(v)$. For a positive integer k, we write $[k]$ for the set $\{1, \ldots, k\}$.

The remainder of the paper is organized as follows. In Section 2, we continue the introductory discussion and present some polynomial preprocessing steps that simplify the input and which we will later on assume performed. We also discuss some relations between the complexities of the problems WIS, EWIS and EWIS$_\alpha$. Section 3 is devoted to the strong **NP**-completeness results. In Section 4, pseudo-polynomial time solutions to the exact weighted independent set problem are presented. We conclude the paper with a short discussion in Section 5 that places the class of line graphs of bipartite graphs between two graph classes with known complexities of the EWIS$_\alpha$ problem.

2 Preliminary observations

The exact weighted independent set problem is (weakly) **NP**-complete for any class of graphs containing $\{\overline{K_n} : n \geq 0\}$. There is a direct equivalence between the exact weighted independent set problem on $\{\overline{K_n} : n \geq 0\}$ and the *subset sum* problem, which is known to be **NP**-complete (see [25]). The subset sum problem is the following: given n integers $a_1, \ldots, a_n$ and a bound B, determine whether there is a subset $J \subseteq [n]$ such that $\sum_{j \in J} a_j = B$.

Hence, we will assume from now on that the weights are encoded in unary. Moreover, if (G, w, M) is an instance of EWIS, we can further assume that $1 \leq w(v) \leq M$, for all $v \in V = V(G)$, as well as $M \leq w(V)$. To see this, we first show that the general case polynomially reduces to solving the problem on n instances with positive weights.

Solving the EWIS problem for any particular instance reduces to solving n problems EWIS$_k$, in which the independent sets are restricted to be of size k, for all $k \in [n]$ (unless $M = 0$, in which case the solution is trivial). The weights in EWIS$_k$ can be assumed to be positive: otherwise, we can add a suitably large constant N to each vertex weight and replace M by $M + kN$ to get an equivalent EWIS$_k$ problem with positive weights only. Finally, applying the same transformation again with $N = w(V) + 1$ reduces the problem EWIS$_k$ to a single EWIS problem with positive weights. Repeating this for all values of $k \in [n]$, the result follows.

Finally, if all vertex weights are positive, we can delete from the graph all vertices whose weight exceeds M, as they will never appear in a solution. Similarly, if $M > w(V)$, then the problem clearly has a negative answer.

The same assumption on vertex weights as for EWIS can also be made for the instances (G, w, M) of its restricted counterpart EWIS$_\alpha$. Again, if some of the weights are negative, we can modify the weights and the target value as we did above for EWIS$_k$. Now we only do it for $k = \alpha(G)$. Note that we can compute $\alpha(G)$ as that only $p \in [n]$ such that the solution to EWIS$_\alpha(G, \mathbf{1}, p)$ is *yes*, where $\mathbf{1}$ denotes the unit vertex weights.

We now discuss some relations between the complexities of the problems WIS, EWIS and EWIS$_\alpha$, when restricted to particular graph classes.

Lemma 2.1. *Let $\mathcal{G}$ be a class of graphs. The following statements are true.*
(i) If EWIS$_\alpha$ *is solvable in pseudo-polynomial time for graphs in $\mathcal{G}$, then* WIS *is solvable in pseudo-polynomial time for graphs in $\mathcal{G}$.*
(ii) If EWIS *is solvable in pseudo-polynomial time for graphs in $\mathcal{G}$, then* EWIS$_\alpha$ *is solvable in pseudo-polynomial time for graphs in $\mathcal{G}$.*
(iii) Let $\mathcal{G}' = \{G' : G \in \mathcal{G}\}$ where $G' = (V', E')$ is the graph, obtained from a graph $G = (V, E) \in \mathcal{G}$, by adding pendant vertices, as follows: $V' = V \cup \{v' : v \in V\}$, $E' =$

$E \cup \{[v, v'] : v \in V\}$. *If* EWIS$_\alpha$ *is solvable in pseudo-polynomial time for graphs in* $\mathcal{G}'$*, then* EWIS *is solvable in pseudo-polynomial time for graphs in* $\mathcal{G}$*.*

Proof. (i) Let (G, w, k) be an instance of the decision version of the weighted independent set problem. As we can assume positive weights, G contains an independent set of total weight at least k if and only if G contains a maximum independent set of total weight at least k. By testing values for M from $w(V)$ down to k and using an algorithm for EWIS$_\alpha$ on the instance (G, w, M), we can decide whether G contains a maximum independent set of total weight at least k.
(ii) Let (G, w, M) be an instance of the EWIS$_\alpha$ problem. It is easy to see that the following algorithm solves EWIS$_\alpha$.
Step 1. Compute $\alpha(G)$, which is equal to the maximum $k \in [n]$ such that the solution to EWIS$(G, \mathbf{1}, k)$ is *yes*, where $\mathbf{1}$ denotes the unit vertex weights.
Step 2. Let $N = w(V) + 1$. For every vertex $v \in V(G)$, let $w'(v) = w(v) + N$. Let $M' = M + \alpha(G)N$. Then it is easy to verify that EWIS$_\alpha(G, w, M) =$ EWIS(G, w', M').
(iii) Let (G, w, M) with $G = (V, E) \in \mathcal{G}$ be an instance of EWIS. Let G' be the graph, defined as in the lemma. Let $n = |V|$ and let $w'(v) = (n+1)w(v)$ for all $v \in V$ and $w'(v) = 1$ for $v \in V'$. Then, it is easy to verify that the solution to EWIS(G, w, M) is *yes* if and only if the solution to EWIS$_\alpha(G', w', M')$ is *yes* for some value M' in the set $\{(n+1)M, \ldots, (n+1)M+n-1\}$. $\square$

The problem EWIS is clearly in **NP**, and so is EWIS$_\alpha$ for any class of graphs $\mathcal{G}$ where IS is polynomially solvable. Therefore, Lemma 2.1 implies the following result.

Corollary 2.2. *Let* $\mathcal{G}$ *be a class of graphs. The following statements are true.*
(i) If WIS *is strongly* ***NP****-complete for graphs in* $\mathcal{G}$*, then* EWIS$_\alpha$ *is strongly* ***NP****-hard for graphs in* $\mathcal{G}$*. If, in addition,* IS *is polynomial for graphs in* $\mathcal{G}$*, then* EWIS$_\alpha$ *is strongly* ***NP****-complete for graphs in* $\mathcal{G}$*.*
(ii) If EWIS$_\alpha$ *is strongly* ***NP****-hard for graphs in* $\mathcal{G}$*, then* EWIS *is strongly* ***NP****-complete for graphs in* $\mathcal{G}$*.*
(iii) Let $\mathcal{G}'$ *be as in Lemma 2.1. If* EWIS *is strongly* ***NP****-complete for graphs in* $\mathcal{G}$*, then* EWIS$_\alpha$ *is strongly* ***NP****-hard for graphs in* $\mathcal{G}'$*. If, in addition,* IS *is polynomial for graphs in* $\mathcal{G}'$*, then* EWIS$_\alpha$ *is strongly* ***NP****-complete for graphs in* $\mathcal{G}'$*.*

Thus, we are mainly interested in determining the complexity (strong **NP**-complete or pseudo-polynomial results) of the exact weighted independent set problem in those classes of graphs where the weighted independent set problem is solvable in pseudo-polynomial time. Moreover, combining parts (ii) and (iii) of the lemma shows that when $\mathcal{G} \in \{$forests, bipartite graphs, chordal graphs$\}$, the problems EWIS and EWIS$_\alpha$ are equivalent (in the sense that, when restricted to the graphs in $\mathcal{G}$, they are either both solvable in pseudo-polynomial time, or they are both strongly **NP**-complete). Recall that a graph G is a *forest* if it is acyclic, *bipartite* if any cycle of G has even length, and *chordal* if any cycle of G with size at least 4 has a chord (i.e., an edge connecting two nonconsecutive vertices of the cycle).

We conclude this subsection by showing that this equivalence remains valid for the class of line graphs. More precisely, if L, $L(Bip)$, $L(K_{2n})$ and $L(K_{n,n})$ denote the classes of line graphs, line graphs of bipartite graphs, line graphs of complete graphs with an even number of vertices, and line graphs of complete balanced bipartite graphs, respectively, we have the following result.

Lemma 2.3. EWIS *is strongly* ***NP****-complete for graphs in* L *(resp.,* $L(Bip)$*) if and only if* EWIS$_\alpha$ *is strongly* ***NP****-complete for graphs in* $L(K_{2n})$ *(resp.,* $L(K_{n,n})$*).*

Proof. The backward implication is given by part (ii) of Lemma 2.1. The forward implication follows from a reduction of the exact matching problem to the exact perfect matching problem which we show now. Given an instance $G = (V, E)$ with edge weights w and a target M for the exact matching problem, construct an instance $(K_{n'}, w', M')$ for the exact perfect matching problem as follows. If $n = |V|$ is odd, we add a new vertex and we complete the graph G. For an edge e of G, let $w'(e) = Nw(e)$ where $N = w(E) + 1$, for an edge $e \notin E$ let $w'(e) = 1$. The transformation is clearly polynomial, and G has a matching of weight M if and only if $K_{n'}$ has a perfect matching of weight $NM + k$ for some value of $k \in \{0, \ldots, n-1\}$. Also, it is easy to see that in the case of bipartite graphs $G = (L, R; E)$ with $|L| \leq |R|$, we can add $|R \setminus L|$ vertices to L to balance the bipartition. □

3 Hardness results

The weighted independent set problem is solvable in polynomial time for bipartite graphs by network flow techniques. However, as we show in this section, the exact version of the problem is strongly **NP**-complete even for cubic bipartite graphs.

3.1 Bipartite graphs

A bipartite graph is a graph $G = (V, E)$ whose vertex set admits a partition $V = L \cup R$ into the left set L and the right set R such that any edge of G links a vertex of L to a vertex of R. In general, a bipartite graph may admit several such partitions. Since we only consider connected bipartite graphs (which have a unique such partition, up to switching the parts), we will also write $G = (L, R; E)$.

The strong **NP**-completeness of EWIS in bipartite graphs is straightforward since the *balanced biclique* problem (also called balanced complete bipartite subgraph) is **NP**-complete [25, 20]. This problem consists in deciding, given a bipartite graph $G = (L, R; E)$ and an integer k, if there exist $L' \subseteq L$ and $R' \subseteq R$ with $|L'| = |R'| = k$ such that the subgraph induced by $L' \cup R'$ is a complete bipartite subgraph (also called *biclique* of size k). In [20], a variation of this latter problem is introduced where we must have $|L'| = a$ and $|R'| = b$ (called the *biclique problem*). From an instance G and k of balanced biclique, we introduce weight 1 on each vertex of L, weight $B = \max\{|L|, |R|\} + 1$ on each vertex of R, and we set $M = k + Bk$. It is clear that there exist an independent set in $(L, R; (L \times R) \setminus E)$ with weight M if and only if there exists a balanced biclique in $(L, R; E)$ of size k.

We now strengthen this result by proving that EWIS_α is strongly **NP**-complete even for cubic bipartite graphs. By contrast, it is easy to see that for graphs of maximum degree 2, EWIS and EWIS_α are pseudo-polynomially solvable problems.[3]

Theorem 3.1. EWIS_α *is strongly* ***NP****-complete in cubic bipartite graphs.*

Proof. The problem is clearly in **NP**, as the IS problem is solvable in polynomial time for bipartite graphs. The hardness reduction is made from the decision version of the clique problem in regular graphs which is known to be **NP**-complete, see [25]. A *clique* V^* is a subset of vertices of G such that the subgraph induced by V^* is complete. Let $G = (V, E)$ be a Δ-regular graph of n vertices and let k be an integer. Without loss of generality, assume that $0 < k < \Delta < n - 1$, since otherwise the problem is easy. We build the instance $I = (G', w)$ of EWIS_α where $G' = (L, R; E')$ is a bipartite graph as follows:

[3] Every connected graph in this class is either a cycle or a path, and the treewidth of such graphs is at most 2. By Corollary 4.2 and Theorem 4.18 from Section 4, the problem is solvable in pseudo-polynomial time in this class.

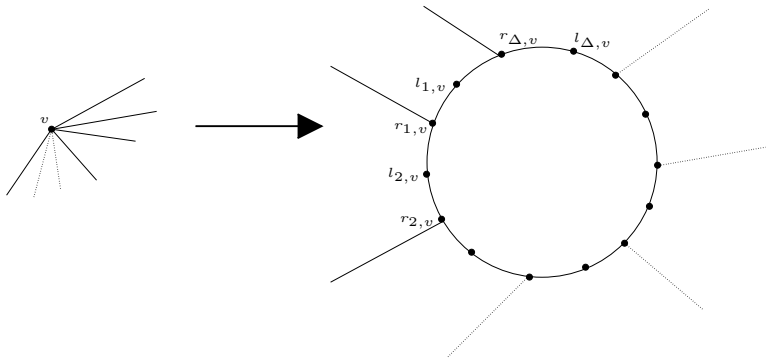

Figure 1: The gadget $H(v)$.

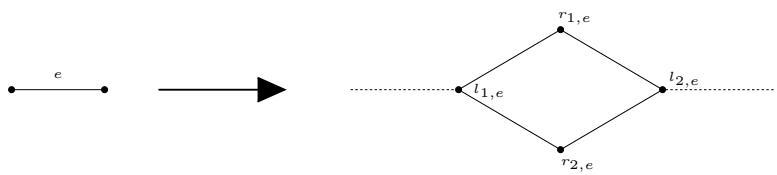

Figure 2: The gadget $H(e)$.

- For each vertex $v \in V$, we construct a gadget $H(v)$ which is a cycle of length 2Δ. Thus, it is a bipartite graph where the left set is $L_v = \{l_{1,v}, \ldots, l_{\Delta,v}\}$ and the right set is $R_v = \{r_{1,v}, \ldots, r_{\Delta,v}\}$. The weights are $w(l_{i,v}) = 1$ and $w(r_{i,v}) = n\Delta(\frac{2+n\Delta}{2})$ for $i \in [\Delta]$. The gadget $H(v)$ is illustrated with Figure 1.

- For each edge $e \in E$, we construct a gadget $H(e)$ which is also a cycle of length 4. Thus, it is a bipartite graph where the left set is $L_e = \{l_{1,e}, l_{2,e}\}$ and the right set is $R_e = \{r_{1,e}, r_{2,e}\}$. The weights are $w(l_{i,e}) = \frac{(n\Delta)}{2}$ and $w(r_{i,e}) = \frac{(n\Delta)^2}{2}(\frac{2+n\Delta}{2})$ for $i = 1, 2$. The gadget $H(e)$ is illustrated with Figure 2.

- We interconnect these gadgets by iteratively applying the following procedure. For each edge $e = [u, v] \in E$, we add two edges $[r_{i,u}, l_{1,e}]$ and $[l_{i,u}, r_{1,e}]$ where $l_{i,u}$ is a neighbor of $r_{i,u}$ in $H(u)$ between gadgets $H(u)$, $H(e)$ and two edges $[r_{j,v}, l_{2,e}]$ and $[l_{j,v}, r_{2,e}]$ where $l_{j,v}$ is a neighbor of $r_{j,v}$ in $H(v)$ between gadgets $H(v)$, $H(e)$ such that the vertices $r_{i,u}$, $l_{i,u}$, $r_{j,v}$ and $l_{j,v}$ have degree 3.

It is clear that G' is bipartite and the weights are polynomially bounded. Moreover, since G is a Δ-regular graph, we conclude that G' is 3-regular.

We claim that there exist a clique V^* of G with size at least k if and only if $\mathrm{EWIS}_\alpha(G', w, M)$ is *yes* with

$$M = k\Delta + n\Delta\frac{k(k-1)}{2} + n\Delta(\frac{2+n\Delta}{2})\left((n-k)\Delta + (\frac{n\Delta}{2} - \frac{k(k-1)}{2})n\Delta\right).$$

Let I be a maximum independent set of G' with $w(I) = M$. Since G' is cubic and bipartite, G' has a perfect matching (for instance, take a perfect matching in each gadget $H(v)$ and $H(e)$), and we conclude that $\alpha(G) = |I| = |R| = |L|$. This implies in particular that for any vertex

$v \in V$, either L_v or R_v is a subset of I. Moreover, the same property holds for any $e \in E$ (i.e., either L_e or R_e is a subset of I). Moreover, by construction of the weights, the quantity $k\Delta$ of M must come from vertices $l_{i,v}$, $r_{i,v}$ or $l_{i,e}$. Since $k < n$, this quantity cannot come from $r_{i,v}$. Moreover, since $l_{i,e} \in I$ if and only if $L_e \subseteq I$, the contribution of L_e in I is $n\Delta$. In this case, the contribution of $k\Delta$ must come from $l_{i,v}$. Thus, we obtain:

$$|I \cap L_V| = k\Delta\,, \qquad |I \cap R_V| = (n-k)\Delta. \tag{1}$$

where $L_V = \cup_{v \in V} L_v$ and $R_V = \cup_{v \in V} R_v$. Thus, using (1) we must obtain:

$$w\left(I \cap (L_E \cup R_E)\right) = n\Delta\frac{k(k-1)}{2} + n\Delta(\frac{2+n\Delta}{2})(\frac{n\Delta}{2} - \frac{k(k-1)}{2})n\Delta. \tag{2}$$

where $L_E = \cup_{e \in E} L_e$ and $R_E = \cup_{e \in E} R_e$. Now, we prove that there are exactly $k(\frac{k-1}{2})$ gadgets $H(e)$ with $L_e \subseteq I$. Assume the reverse; then, $|I \cap L_E| = k(k-1) - 2p$ and $|I \cap R_E| = n\Delta - k(k-1) + 2p$ for some $p \neq 0$ (p can be negative). Combining these equalities with equality (2), we deduce that $p = 0$, contradiction.

Thus, if we set $V^* = \{v \in V : L_v \subseteq I\}$, we deduce from previously $|V^*| = k$ and we will have necessarily that V^* is a clique of G.

Conversely, let V' be a clique of G with $|V'| \geq k$ and consider a subclique $V^* \subseteq V'$ of size exactly k. We set $S = S_L \cup S_R$ with $S_L = \cup_{v \in V^*} L_v \cup_{e \in E(V^*)} L_e$ and $S_R = \cup_{v \in V \setminus V^*} R_v \cup_{e \in E \setminus E(V^*)} R_e$. One can easily verify that $w(I) = M$ and that I is a maximum independent set of G'. Indeed, let us assume the converse; thus, there exist $r_{i,v} \in I$ (and thus $R_v \subseteq I$), $l_{j,e} \in I$ (with $j = 1, 2$) and $[r_{i,v}, l_{j,e}] \in E'$. By construction of I, we deduce that $e = [u, v] \in E(V^*)$ and then $L_v \subseteq I$, contradiction. The proof is complete. □

As corollary of Theorem 3.1, we can derive that the biclique problem remains **NP**-complete when the minimum degree of $G = (L, R; E)$ is $n - 3$ where $|L| = |R| = n$. In this case, we replace any gadget $H(e)$ of Theorem 3.1 by a cycle of length $2n\Delta$ and we delete edges $[l_{i,u}, r_{1,e}]$ and $[l_{j,v}, r_{2,e}]$.

We also remark that Theorem 3.1 implies the strong **NP**-completeness of EWIS$_\alpha$ for perfect graphs, one of the well-known classes where WIS is solvable in polynomial time.

3.2 A more general hardness result

In this section, we are going to strengthen the result of Theorem 3.1 to a more general setting: for hereditary subclasses of bipartite graphs in which no vertex degree exceeds 3.[4] To this end, we first introduce some notations. We will denote the class of graphs containing no induced subgraphs from a set $\mathcal{F}$ by *Free*($\mathcal{F}$). Any graph in *Free*($\mathcal{F}$) will be called $\mathcal{F}$-free. Our hardness results will be expressed in terms of a parameter related to the set of forbidden induced subgraphs $\mathcal{F}$.

For $i \geq 1$, let H_i denote the graph depicted on Figure 3 and for $i \geq 3$, let C_i denote the chordless cycle of length i. We associate to every graph G a parameter $\kappa(G)$. If G has a vertex of degree 4 or more, we define $\kappa(G)$ to be 0. Otherwise, let $\kappa(G)$ denote the minimum value of $k \geq 3$ such that G contains an induced copy of either C_k or H_k (or ∞ if no such k exists, and with the obvious convention that no graph contains an induced C_1 or C_2). Also, for a (possibly infinite) set of graphs $\mathcal{F}$, we define $\kappa(\mathcal{F}) = \sup\{\kappa(G) \ : \ G \in \mathcal{F}\}$.

With these definitions in mind, we can use the strong **NP**-hardness of EWIS$_\alpha$ for bipartite graphs of degree at most 3, and the reduction typically used for the independent set problem (see e.g. [39, 43]), to derive the following hardness result.

[4] A class of graphs is *hereditary* if it is closed under vertex deletion.

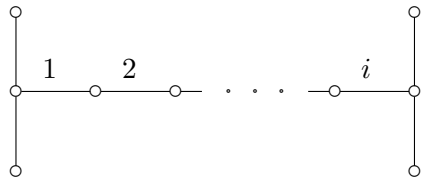

Figure 3: The graph H_i

Theorem 3.2. *Let $\mathcal{G}$ be the class of $\mathcal{F}$-free bipartite graphs of maximum degree at most 3. If $\kappa(\mathcal{F}) < \infty$, then* EWIS_α *is strongly* ***NP****-complete in the class $\mathcal{G}$.*

Proof. The problem is clearly in **NP**. We show completeness in two steps. First, for $k \geq 3$, let $\mathcal{S}_k$ be the class of all bipartite $(C_3, \ldots, C_k, H_1, \ldots, H_k)$-free graphs of vertex degree at most 3, and let us show that for any fixed k, the problem is strongly **NP**-complete for graphs in $\mathcal{S}_k$. Let (G, w, M) be an instance of EWIS_α where G is a bipartite graph of maximum degree at most 3.

We can transform the graph G in polynomial time to a weighted graph G', as follows. Let $k' = \lceil \frac{k}{2} \rceil$. We replace each edge e of G by a path $P(e)$ on $2k' + 2$ vertices. Let $N = w(V) + 1$. We set the weights w' of the endpoints of $P(e)$ equal to the weights of the corresponding endpoints of e, while each internal vertex of $P(e)$ gets weight N. It is easy to verify that G' belongs to $\mathcal{S}_k$.

We claim that the answer to $\mathrm{EWIS}_\alpha(G, w, M)$ is *yes* if and only if the answer to $\mathrm{EWIS}_\alpha(G', w', M + mk'N)$ is *yes*, where $m = |E(G)|$.

One direction is immediate, as each maximum independent set of G can be extended to a maximum independent set of G', by simply adding k' internal vertices of each newly added path. Doing so, the weight increases by $mk'N$.

Suppose now that the answer to $\mathrm{EWIS}_\alpha(G', w', M + mk'N)$ is *yes*. Let I' be a maximum independent set of G' of weight $M + mk'N$. Since I' is independent, it can contain at most k' internal vertices of each newly added path. Therefore, for each $e \in E(G)$, the set I' must contain *exactly* k' internal vertices of $P(e)$ – otherwise its weight would be at most $W + (mk' - 1)N$, contradicting our choice of N.

Let I denote the set, obtained from I' by deleting the internal vertices of newly added paths. Then, I is an independent set of G. Indeed, if $e = [u, v] \in E(G)$ for some $u, v \in I$, then I' can contain at most $k' - 1$ internal vertices of $P(e)$, contradicting the above observation. Also, it is easy to see that I is a maximum independent set of G. Finally, as the weight of I is exactly M, we conclude that the answer to $\mathrm{EWIS}_\alpha(G, w, M)$ is *yes*.

This shows that EWIS_α is strongly **NP**-complete in the class $\mathcal{S}_k$. To prove strong **NP**-completeness of the problem in the class $\mathcal{G}$, we now show that the class $\mathcal{G}$ contains all graphs in $\mathcal{S}_k$, for $k := \max\{3, \kappa(\mathcal{F})\}$. Let G be a graph from $\mathcal{S}_k$. Assume that G does not belong to $\mathcal{G}$. Then G contains a graph $A \in \mathcal{F}$ as an induced subgraph. From the choice of G we know that A belongs to $\mathcal{S}_k$, but then $k < \kappa(A) \leq \kappa(\mathcal{F}) \leq k$, a contradiction. Therefore, $G \in \mathcal{G}$ and the theorem is proved. $\square$

4 Polynomial results

In this section, we present pseudo-polynomial solutions to the exact weighted independent set problem, when the input graphs are restricted to particular classes.[5] Our algorithms mostly resemble those for the WIS problem in respective graph classes, and are based either on a dynamic programming approach (Section 4.1), or on modular decomposition (Section 4.2). Often, the solutions will be described using the following pseudo-polynomially equivalent version of EWIS: given a weighted graph and a target M, compute all M instances of EWIS with varying target. That is, the output is given by the vector $(\text{EWIS}(G, w, k) : k \in [M])$ in $\{yes, no\}^{[M]}$.

First, we show that without loss of generality, we may restrict our attention to connected graphs. To do so, we recall the *subset sum* problem, a typical example of an **NP**-complete problem that can be solved in pseudo-polynomial time by dynamic programming.

SUBSET SUM
INSTANCE: Positive integers $a_1, \ldots, a_n, b$.
QUESTION: Is there a subset S of $[n]$ such that $\sum_{i \in S} a_i = b$?

We consider the following generalization of the subset sum problem.

GENERALIZED SUBSET SUM (GSS)
INSTANCE: Nonempty sets of positive integers $A_1, \ldots, A_n$ and a positive integer b.
QUESTION: Is there a nonempty subset S of $[n]$ and a mapping $a : S \to \cup_{i \in S} A_i$ such that $a(i) \in A_i$ for all $i \in S$, and $\sum_{i \in S} a(i) = b$?

It is straightforward to extend the dynamic programming solution for subset sum to one for *generalized subset sum*.

Lemma 4.1. *Generalized subset sum can be solved in time $O(nb^2)$ by dynamic programming.*

Proof. We are going to show a stronger statement: in the stated time, not only we can verify if there is an S and an a such that $\sum_{i \in S} a(i) = b$ for the given b, but we will generate the set B of *all* values $b' \in [b]$ with such property.

The proof is by induction on n. The statement clearly holds for $n = 1$, as in this case we have $B = \{b' \in A_1 : b' \leq b\}$.

Suppose now that $n > 1$. Let $I = (A_1, \ldots, A_n; b)$ be an instance of the GSS problem. Let B' be the inductively constructed set of all possible values of $b' \in [b]$ such that the solution to the GSS problem on the instance $(A_1, \ldots, A_{n-1}; b')$ is *yes*. By induction, the set B' was constructed in time $O((n-1)b^2)$.

Let $\beta \in [b]$. Then, β will belong to B, i.e., the solution to the GSS, given $(A_1, \ldots, A_n; \beta)$, will be *yes*, if and only if either $\beta \in B'$, or we can write β as $\beta = b' + a_n$ for some $b' \in B'$ and $a_n \in A_n$. In other words, $B = B' \cup B''$, where B'' denotes the set of all such sums: $B'' = \{b' + a_n : b' \in B', a_n \in A_n, b' + a_n \leq b\}$.

The set B'' can be constructed in time $O(b^2)$. Adding this time complexity to the time $O((n-1)b^2)$ needed to construct B' proves the lemma. □

It now follows immediately that in order to solve EWIS, it suffices to solve the problem for connected graphs.

[5] For a comprehensive survey on graph classes, see [13].

Corollary 4.2. *Let (G, w, M) be an instance of* EWIS*, and let $C_1, \ldots, C_r$ be the connected components of G. Suppose that for each $i \in [r]$, the set of solutions* $(\mathrm{EWIS}(C_i, w, k) : k \in [M])$ *for C_i is given. Then, we can compute the set of solutions* $(\mathrm{EWIS}(G, w, k) : k \in [M])$ *for G in time $O(rM^2)$.*

Proof. It suffices to observe that for every $k \in [M]$, the solution to $\mathrm{EWIS}(G, w, k)$ is *yes* if and only if the solution to the GSS problem on the instance $(A_1, \ldots, A_r; k)$ is *yes*, where A_i denotes the set of all values $k' \in [M]$ such that the solution to $\mathrm{EWIS}(C_i, w, k')$ is *yes*. □

4.1 Dynamic programming solutions

We can summarize the results of this subsection in the following theorem.

Theorem 4.3. *The problems* EWIS *and* EWIS_α *admit pseudo-polynomial-time solutions in each of the following graph classes: mK_2-free graphs, interval graphs and their generalizations k-thin graphs, chordal graphs, AT-free graphs, $(claw, net)$-free graphs, distance-hereditary graphs, circle graphs, graphs of treewidth at most k, and graphs of clique-width at most k.*

The rest of this subsection is devoted to proving this result. By part (ii) of Lemma 2.1, it suffices to develop pseudo-polynomial solutions for EWIS. All graph classes mentioned in the theorem are defined in the sequel. Most of the algorithms resemble those for the WIS problem and exploit the special structure of graphs in the classes. Thus, we do not present all of the algorithms and their analyses in full detail; for more details, we refer the reader to the technical report [35].

4.1.1 mK_2-free graphs

Our first example deals with graphs with no large induced matchings. Recall that K_2 denotes the graph consisting of two adjacent vertices. The disjoint union of m copies of K_2 is denoted by mK_2. Thus, graphs whose largest induced matching consists of less than m edges are precisely the mK_2-free graphs.

Theorem 4.4. *For every positive integer m,* EWIS *admits a pseudo-polynomial algorithm for mK_2-free graphs.*

Proof. All maximal independent sets $I_1, \ldots, I_N$ in an mK_2-free graph can be found in polynomial time [1, 3, 44, 50]. Since every independent set is contained in a maximal one, $\mathrm{EWIS}(G, w, k)$ will take the value *yes* if and only if there is an $i \in [N]$ such that $\mathrm{EWIS}(G[I_i], w, k)$ is *yes*. Thus, the EWIS problem in mK_2-free graphs reduces to solving polynomially many instances of the subset sum problem. □

4.1.2 Interval graphs

Interval graphs are one of the most natural and well-understood classes of intersection graphs. They are intersection graphs of intervals on the real line, and many optimization problems can be solved by dynamic programming on these graphs.

Formally, given a collection $\mathcal{I} = ([a_i, b_i] : i \in I)$ of intervals on the real line, its *intersection graph* $G(\mathcal{I})$ is defined by $V(G(\mathcal{I})) = \mathcal{I}$, and there is an edge connecting $[a_i, b_i]$ and $[a_j, b_j]$ if and only if $[a_i, b_i] \cap [a_j, b_j] \neq \emptyset$. The collection $\mathcal{I}$ is said to be an *interval model* of $G(\mathcal{I})$. A graph G is said to be an *interval graph* if it admits an interval model, i.e., if there is a collection $\mathcal{I}$ of intervals on the real line such that $G = G(\mathcal{I})$.

A representation of interval graphs that is particularly suitable for the EWIS problem is the following. It has been shown by Ramalingam and Pandu Rangan [45] that a graph $G = (V, E)$ is interval if and only if it admits a vertex ordering $(v_1, \ldots, v_n)$ such that for all triples (r, s, t) with $1 \leq r < s < t \leq n$, the following implication is true:

$$\text{if} \quad [v_r, v_t] \in E \quad \text{then} \quad [v_s, v_t] \in E\,.$$

Moreover, such an ordering of an interval graph can be found in time $O(n + m)$. Based on this ordering, we can prove the following statement.

Theorem 4.5. EWIS *admits an* $O(Mn + m)$ *algorithm for interval graphs.*

Proof. Let $(v_1, \ldots, v_n)$ be a vertex ordering such that $[v_s, v_t] \in E$, whenever $[v_r, v_t] \in E$, for all triples (r, s, t) with $1 \leq r < s < t \leq n$.

For every $i \in [n]$, let G_i denote the subgraph of G induced by $\{v_1, \ldots, v_i\}$ (also, let G_0 be the empty graph). Then, for every $i \in [n]$, either there is a $j = j(i)$ such that $N_{G_i}(v_i) = \{j, j+1, \ldots, i-1\}$, or $N_{G_i}(v_i) = \emptyset$ (in which case let us define $j(i) = i$). Now, if I is an independent set of G_i, then either $v_i \in I$ (in which case $I \backslash \{v_i\}$ is an independent set of $G_{j(i)-1}$), or $v_i \notin I$ (in which case I is an independent set of G_{i-1}). This observation is the key to the following simple $O(Mn + m)$ dynamic programming solution to the EWIS problem on interval graphs.

Step 1. Find a vertex ordering $(v_1, \ldots, v_n)$ as above.
Step 2. Set EWIS(G_0, w, k) to *no* for all $k \in [M]$.
Step 3. For $i = 1, \ldots, n$, do the following:

3.1. Find $j \in [i]$ such that $N_{G_i}(v_i) = \{j, j+1, \ldots, i-1\}$.

3.2. For $k \in [M]$, do the following:
If $k = w(v_i)$, set EWIS(G_i, w, k) to *yes*.
If $k < w(v_i)$, set EWIS(G_i, w, k) to EWIS(G_{i-1}, w, k).
If $k > w(v_i)$, set EWIS(G_i, w, k) to *yes* if at least one of the solutions to EWIS$(G_{j(i)-1}, w, k - w(v_i))$ and EWIS(G_{i-1}, w, k) is *yes*, and to *no* otherwise.

Step 4. Output the solution to EWIS(G_n, w, M). □

4.1.3 k-thin graphs

The property used in the above characterization of interval graphs has been generalized by Mannino *et al.* in [33], where they define the class of k-thin graphs. A graph $G = (V, E)$ is said to be *k-thin* if there exist an ordering $(v_1, \ldots, v_n)$ of V and a partition of V into k classes such that, for each triple (r, s, t) with $1 \leq r < s < t \leq n$, if v_r, v_s belong to the same class and $[v_r, v_t] \in E$, then $[v_s, v_t] \in E$.

Let us mention at this point that finding a feasible frequency assignment of a given cost can be modeled as the EWIS problem on a k-thin graph, where the parameter k depends on the input to the frequency assignment problem. For further details, we refer the reader to the paper [33].

Based on the same idea as for interval graphs, a dynamic programming solution for k-thin graphs can be obtained, provided we are given an ordering and a partition of the vertex set.

Theorem 4.6. *Suppose that for a k-thin graph $G = (V, E)$, $k \geq 2$, an ordering $(v_1, \ldots, v_n)$ of V and a partition of V into k classes are given such that, for each triple (r, s, t) with $1 \leq r < s < t \leq n$, if v_r, v_s belong to the same class and $[v_r, v_t] \in E$, then $[v_s, v_t] \in E$. Then,* EWIS *admits an $O(Mn^k)$ algorithm for G.*

Proof. The proof is a straightforward extension of the proof of Theorem 4.5. Let $V_1, \ldots, V_k$ be the classes of the partition. Instead of the graphs G_i, induced by the first i vertices, we now consider all graphs $G(i_1, \ldots, i_k)$, induced by the "first" i_r vertices of each class (according to the ordering on V restricted to the class), for all $r \in \{1, \ldots, k\}$, and for all $O(n^k)$ choices of such k-tuples $(i_1, \ldots, i_k) \in \{1, \ldots, |V_1|\} \times \ldots \times \{1, \ldots, |V_k|\}$. □

4.1.4 Chordal graphs

Chordal (or triangulated) graphs are graphs in which every cycle of length at least four has a chord. They strictly generalize interval graphs and provide another class where the WIS problem is polynomially solvable. Unfortunately for our purpose, the usual approaches for the WIS problem in chordal graphs ([23, 48]) heavily rely on the maximization nature of the problem, and generally do not preserve the overall structure of independent sets. As such, they do not seem to be directly extendable to the exact version of the problem. Instead, we develop a pseudo-polynomial time solution to the EWIS problem in chordal graphs by using one of the many different characterizations of chordal graphs: their *clique tree representation.*

Theorem 4.7. EWIS *admits an* $O(M^2 n(n+m))$ *algorithm for chordal graphs.*

Proof. Given a chordal graph G, we first compute a *clique tree* of G. This can be done in time $O(n+m)$ [29]. A clique tree of a chordal graph G is a tree T whose nodes are the maximal cliques of G, such that for every vertex v of G, the subgraph T_v of T induced by the maximal cliques containing v is a tree. Furthermore, we fix an arbitrary node K_r in the clique tree in order to obtain a *rooted clique tree.* For a maximal clique K, we denote by $G(K)$ the subgraph of G induced by the vertices of K and all vertices contained in some descendant of K in T.

The algorithm is based on a set of identities developed by Okamoto, Uno and Uehara in [40], where a clique tree representation was used to develop linear-time algorithms to count independent sets in a chordal graph. Let $\mathcal{IS}(G)$ be the family of independent sets in G. For a vertex v, let $\mathcal{IS}(G, v)$ be the family of independent sets in G that contain v. For a vertex set U, let $\overline{\mathcal{IS}}(G, U)$ be the family of independent sets in G that contain no vertex of U. Consider a maximal clique K of G, and let $K_1, \ldots, K_l$ be the children of K in T. (If K is a leaf of the clique tree, we set $l = 0$.) Then, as shown in [40], for every distinct $i, j \in [l]$, the sets $V(G(K_i)) \backslash K$ and $V(G(K_j)) \backslash K$ are disjoint. Moreover, if $\sqcup$ denotes the disjoint union, the following relations hold:

$$\begin{aligned}
\mathcal{IS}(G(K)) &= \overline{\mathcal{IS}}(G(K), K) \sqcup \textstyle\bigsqcup_{v \in K} \mathcal{IS}(G(K), v)\,; \\
\mathcal{IS}(G(K), v) &= \left\{ I \cup \{v\} \;\middle|\; I = \textstyle\bigcup_{i=1}^{l} I_i,\ I_i \in \left\{ \begin{array}{ll} \mathcal{IS}(G(K_i), v), & \text{if } v \in K_i; \\ \overline{\mathcal{IS}}(G(K_i), K \cap K_i), & \text{otherwise.} \end{array} \right\} \right. ; \\
\overline{\mathcal{IS}}(G(K), K) &= \left\{ I \;\middle|\; I = \textstyle\bigsqcup_{i=1}^{l} I_i,\ I_i \in \overline{\mathcal{IS}}(G(K_i), K \cap K_i) \right\}\,; \\
\overline{\mathcal{IS}}(G(K_i), K \cap K_i) &= \overline{\mathcal{IS}}(G(K_i), K_i)) \sqcup \textstyle\bigsqcup_{u \in K_i \backslash K} \mathcal{IS}(G(K_i), u) \quad \text{for each } i \in [l]\,.
\end{aligned}$$

We extend our usual Boolean predicate $\mathrm{EWIS}(H, w, k)$ to the following two: for a vertex v of a weighted graph (H, w) and in integer k, let $\mathrm{EWIS}(H, w, k, v)$ denote the Boolean predicate that is *yes* if and only if in H there is an independent set I of total weight k that contains v. Also, for a set of vertices U let $\overline{\mathrm{EWIS}}(H, w, k, U)$ take the value *yes* if and only if in H there is an independent set of total weight k that contains no vertex from U. Based on the above equations, we can develop the following recursive relations for EWIS:

$$\mathrm{EWIS}(G(K), w, k) = \overline{\mathrm{EWIS}}(G(K), w, k, K) \vee \bigvee_{v \in K : w(v) \leq k} \mathrm{EWIS}(G(K), w, k, v) \tag{3}$$

where $\vee$ denotes the usual *Boolean OR function* (with the obvious identification *yes* $\leftrightarrow 1$, *no* $\leftrightarrow 0$). That is, its value is *yes* if at least one of its arguments is *yes*, and *no* otherwise.

$$\mathrm{EWIS}(G(K), w, k, v) = \mathbf{GSS}(A_1, \ldots, A_l, k - w(v)) \tag{4}$$

where $\mathbf{GSS}(A_1, \ldots, A_l, k)$ denotes the solution to the generalized subset sum problem with the input $(A_1, \ldots, A_l, k)$, and the sets A_i for $i \in [l]$ are given by

$$A_i = \begin{cases} \{k' - w(v) \ : \ w(v) \leq k' \leq k, \ \mathrm{EWIS}(G(K_i), w, k', v) = yes\}, & \text{if } v \in K_i; \\ \{k' \ : \ 1 \leq k' \leq k, \ \overline{\mathrm{EWIS}}(G(K_i), w, k', K \cap K_i) = yes\}, & \text{otherwise.} \end{cases}$$

Note that if $I_i \in \mathcal{IS}(G(K_i), v)$ and $I_j \in \mathcal{IS}(G(K_j), v)$ for some distinct indices $i, j \in [l]$, then we have $I_i \cap I_j = \{v\}$. Moreover, since this is the only possible nonempty intersection of two independent sets from $\bigcup_{i=1}^{l} I_i$ in the equation for $\mathcal{IS}(G(K), v)$, it follows that the sum of the weights of the sets $I_i \backslash \{v\}$ (over all $i \in [l]$) equals to the weight of $\left(\bigcup_{i=1}^{l} I_i\right) \backslash \{v\}$, thus justifying Equation (4).

Similarly, we have

$$\overline{\mathrm{EWIS}}(G(K), w, k, K) = \mathbf{GSS}(A_1, \ldots, A_l, k) \tag{5}$$

where, for each $i \in [l]$, the set A_i is given by

$$A_i = \{k' : 1 \leq k' \leq k, \ \overline{\mathrm{EWIS}}(G(K_i), w, k', K \cap K_i) = yes\},$$

and, finally, for each $i \in [l]$, we have:

$$\overline{\mathrm{EWIS}}(G(K_i), w, k, K \cap K_i) = \overline{\mathrm{EWIS}}(G(K_i, w, k, K_i)) \vee \bigvee_{u \in K_i \backslash K} \mathrm{EWIS}(G(K_i), w, k, u). \tag{6}$$

Given the above equations, it is now easy to develop a pseudo-polynomial dynamic programming algorithm. Having constructed a rooted tree T of G, we traverse it in a bottom-up manner. For a leaf K, we have

$$\overline{\mathrm{EWIS}}(G(K), w, k, K) = \begin{cases} yes, & \text{if } k = 0; \\ no, & \text{otherwise.} \end{cases}$$

and

$$\mathrm{EWIS}(G(K), w, k, v) = \begin{cases} yes, & \text{if } w(v) = k; \\ no, & \text{otherwise.} \end{cases}.$$

For every other node K, we compute the values of $\overline{\mathrm{EWIS}}(G(K), w, k, K)$ and $\mathrm{EWIS}(G(K), w, k, v)$ by referring to the recursive relations (6), (5) and (4) in this order. Finally, the value of $\mathrm{EWIS}(G, w, k)$ is given by $\mathrm{EWIS}(G(K_r), w, k)$, which can be computed using Equation (3).

The correctness of the procedure follows immediately from the above discussion. To justify the time complexity, observe that in a node K of the tree with children $K_1, \ldots, K_l$, the number of operations performed is $O(\sum_{i=1}^{l} |K_i| + lM^2 + |K| lM^2)$. Summing up over all the nodes of the clique tree, and using the fact that a chordal graph has at most n maximal cliques, which satisfy $\sum_{K \in V(T)} |K| = O(n + m)$, the claimed complexity bound follows. □

4.1.5 AT-free graphs

The class of *AT-free graphs* is another class that contains the interval graphs. Moreover, AT-free graphs contain other well-known subclasses of perfect graphs, for instance *permutation graphs* and their superclass, the class of *co-comparability graphs*.

A triple $\{x, y, z\}$ of pairwise non-adjacent vertices in a graph G is an *asteroidal triple* if for every two of these vertices there is a path between them avoiding the closed neighborhood of the third. Formally, x and y are in the same component of $G - N[z]$, x and z are in the same component of $G - N[y]$, and y and z are in the same component of $G - N[y]$.[6] A graph is called *AT-free* if it has no asteroidal triples.

Our dynamic programming algorithm that solves EWIS for AT-free graphs is based on the dynamic programming approach to the WIS problem in AT-free graphs, developed by Broersma, Kloks, Kratsch and Müller in [16]. Let us start with a definition.

Definition 1. *Let x and y be two distinct nonadjacent vertices of an AT-free graph G. The interval $I(x, y)$ is the set of all vertices z of $V(G)\backslash\{x, y\}$ such that x and z are in one component of $G - N[y]$, and z and y are in one component of $G - N[x]$.*

Now, we recall some structural results from [16].

Theorem 4.8 ([16])**.** *Let $I = I(x, y)$ be a nonempty interval of an AT-free graph G, and let $s \in I$. Then there exist components $C_1^s, \ldots, C_t^s$ of $G - N[s]$ such that the components of $I\backslash N[s]$ are precisely $I(x, s)$, $I(s, y)$, and $C_1^s, \ldots, C_t^s$.*

Theorem 4.9 ([16])**.** *Let G be an AT-free graph, let C be a component of $G - N[x]$, let $y \in C$, and let D be a component of the graph $C - N[y]$. Then $N[D] \cap (N[x]\backslash N[y]) = \emptyset$ if and only if D is a component of $G - N[y]$.*

Theorem 4.10 ([16])**.** *Let G be an AT-free graph, let C be a component of $G - N[x]$, let $y \in C$, and let C' be the component of $G - N[y]$ that contains x. Let $B_1, \ldots, B_l$ denote the components of the graph $C - N[y]$ that are contained in C'. Then $I(x, y) = \cup_{i=1}^{l} B_i$.*

We will also need the following general statement.

Observation 4.11. *Let (G, w) be a weighted graph. Then, the solution to* $\mathrm{EWIS}(G, w, k)$ *is yes if and only if there is a vertex $x \in V(G)$ such that the solution to* $\mathrm{EWIS}(G - N(x), w, k)$ *is yes.*

Combining this observation with Theorems 4.9 and 4.10, we obtain the following lemma.

Lemma 4.12. *Let (G, w) be a weighted AT-free graph, $G = (V, E)$. Let $x \in V$ and let C be a component of $G - N[x]$. For a vertex y of C, let C_y denote the subgraph of G induced by $C - N(y)$. Then, the solution to* $\mathrm{EWIS}(C, w, k)$ *is yes if and only if there is a vertex $y \in C$ such that the solution to* $\mathrm{EWIS}(C_y, w, k)$ *is yes. Moreover, the connected components of such a C_y are precisely $\{y\}$, $I(x, y)$, and the components of $G - N[y]$ contained in C.*

Similarly, using Theorem 4.8 we obtain the following conclusion.

Lemma 4.13. *Let (G, w) be a weighted AT-free graph, $G = (V, E)$. Let $I = I(x, y)$ be an interval of G. If $I = \emptyset$, then the solution to EWIS$(G[I], w, k)$ is yes if and only if $k = 0$. Otherwise, let us denote by I_s the subgraph of G induced by $I - N(s)$, for all $s \in I$. Then, the solution to* $\mathrm{EWIS}(I, w, k)$ *is yes if and only if there is a vertex $s \in I$ such that the solution to* $\mathrm{EWIS}(I_s, w, k)$ *is yes. Moreover, the connected components of such an I_s are precisely $\{s\}$, $I(x, s)$, $I(s, y)$, and the components of $G - N[s]$ contained in I.*

[6] Recall that the closed neighborhood of x is defined as $N[x] = N(x) \cup \{x\}$.

Theorem 4.14. EWIS *admits a pseudo-polynomial algorithm for AT-free graphs.*

Proof. It follows from the above discussion that the following pseudo-polynomial algorithm correctly solves the problem.

Step 1. For every $x \in V$ compute all components of $G - N[x]$.
Step 2. For every pair of nonadjacent vertices $x, y \in V(G)$ compute the interval $I(x, y)$.
Step 3. Sort all the components and intervals according to nonincreasing number of vertices.
Step 4. In the order of Step 3, compute the solutions to EWIS(C, w, k), for each component C (for all $k \in \{0, 1, \ldots, w(C)\}$) and the solutions to EWIS(I, w, k) for each interval I (for all $k \in \{0, 1, \ldots, w(I)\}$). To compute the solutions to EWIS(C, w, k) for a component C, first compute the solutions to EWIS$(C - N(y), w, k)$, for all $y \in C$, by applying Lemma 4.12 and Corollary 4.2. Similarly, to compute the solutions to EWIS(I, w, k) for an interval I, first compute the solutions to EWIS$(I - N(s), w, k)$, for all $s \in I$, by applying Lemma 4.13 and Corollary 4.2.
Step 5. Compute the solution to EWIS(G, w, k) using Observation 4.11 and Corollary 4.2. □

A *claw* is the graph $K_{1,3}$. A *net* is the graph obtained from a triangle by attaching one pendant edge to each vertex. In [10], it is shown that for every vertex v of a $(claw, net)$-free graph G, the non-neighborhood of v in G is AT-free. Thus, Theorem 4.14 immediately implies the following result.

Corollary 4.15. EWIS *admits a pseudo-polynomial algorithm for* $(claw, net)$*-free graphs.*

4.1.6 Distance hereditary graphs

A graph is *distance-hereditary* if the distance between any two connected vertices (that is, vertices in the same connected component) is the same in every induced subgraph in which they remain connected.[7] Bandelt and Mulder provided in [4] a *pruning sequence* characterization of distance-hereditary graphs: whenever a graph contains a vertex of degree one, or a vertex with a twin (another vertex sharing the same neighbors), remove such a vertex. A graph is distance-hereditary if and only if it the application of such vertex removals results in a single-vertex graph.

A pruning sequence of a distance-hereditary graph can be computed in linear time [19] and can be useful for algorithmic developments on distance-hereditary graphs. A solution to the WIS problem in distance-hereditary graphs based on the pruning sequence characterization has been developed by Cogis and Thierry in [17]. It turns out that their approach can be generalized in order to solve the exact version of the problem (a detailed description can be found in [35]).

Theorem 4.16. EWIS *admits an* $O(M^2n + m)$ *algorithm for distance-hereditary graphs.*

4.1.7 Circle graphs

Besides intervals on the real line, chords on a circle provide another popular intersection model. The intersection graphs of chords on a circle are called *circle graphs*. In this subsection, we present a $O(M^2n^2)$ dynamic-programming algorithm for the EWIS problem in circle graphs. Our algorithm for EWIS on circle graphs is based on the dynamic programming solution for the IS problem, developed by Supowit in [47].

[7] The distance between two vertices u and v in a connected graph G is the length (i.e., the number of edges) of a shortest path connecting them.

Theorem 4.17. EWIS *admits an* $O(M^2n^2)$ *algorithm for circle graphs.*

Proof. Consider a finite set of N chords on a circle. We may assume without loss of generality that no two chords share an endpoint. Number the endpoints of the chords from 1 to $2N$ in the order as they appear as we move clockwise around the circle (from an arbitrary but fixed starting point).

The idea is simple. For $1 \leq i < j \leq 2N$, let $G(i,j)$ denote the subgraph of G induced by chords whose both endpoints belong to the set $\{i, i+1, \ldots, j\}$. Obviously $G = G(1, 2N)$.

Let $1 \leq i < j \leq 2N$. If $j = i+1$ then the solution to EWIS$(G(i,j), w, k)$ is *yes* if and only if either $k = 0$, or $(i, i+1)$ is a chord and $k = w((i, i+1))$.

Otherwise, let r be the other endpoint of the chord whose one endpoint is j. If $r < i$ or $r > j$, then no independent set of the graph $G(i,j)$ contains the chord (r,j), so the solution to EWIS$(G(i,j), w, k)$ is *yes* if and only if the solution to EWIS$(G(i,j-1), w, k)$ is *yes*. Suppose now that $i \leq r \leq j-1$ and let I be an independent set of $G(i,j)$. The set I may or may not contain the chord (r,j). If I does not contain (r,j), then I is an independent set of of $G(i,j-1)$ as well. If I contains (r,j), then no other chord in I can intersect the chord (r,j). In particular, this implies that I is of the form $I = \{(r,j)\} \cup I_1 \cup I_2$ where I_1 is an independent set of $G(i, r-1)$ and I_2 is an independent set of $G(r+1, j-1)$.

Therefore, the solution to EWIS$(G(i,j), w, k)$ is *yes* if and only if either the solution to EWIS$(G(i,j-1), w, k)$ is *yes*, or the solution to EWIS(G', w, k) is *yes*, where G' is the graph whose connected components are $G[\{(r,j)\}]$, $G(i, r-1)$ and $G(r+1, j-1)$. Assuming that the solutions for $G(i, r-1)$ and $G(r+1, j-1)$ have already been obtained recursively, we can apply Corollary 4.2 in this case.

The above discussion implies an obvious $O(M^2n^2)$ algorithm that correctly solves the problem. □

4.1.8 Graphs of treewidth at most k

Graphs of treewidth at most k, also known as *partial* k*-trees*, are very important from an algorithmic viewpoint: many graph problems that are **NP**-hard for general graphs are solvable in linear time when restricted to graphs of treewidth at most k [2].[8] A standard dynamic programming approach shows that this is also the case for EWIS.

Theorem 4.18. *For every fixed* k*,* EWIS *admits an* $O(M^2n)$ *algorithm for graphs of treewidth at most* k*.*

A detailed description of the algorithm and its analysis can be found in [35].

4.1.9 Graphs of clique-width at most k

The *clique-width* of a graph G is defined as the minimum number of labels needed to construct G, using the following four graph operations:

(*i*) Create a new vertex v with label i (denoted by $i(v)$).

(*ii*) Take the disjoint union of two labeled graphs G and H (denoted by $G \oplus H$).

(*iii*) Join by an edge each vertex with label i to each vertex with label j ($i \neq j$, denoted by $\eta_{i,j}$).

(*iv*) Rename label i to j (denoted by $\rho_{i \to j}$).

[8] For a definition of treewidth, see e.g. [13].

An expression built from the above four operations is called a *clique-width expression*. A clique-width expression using k labels is called a *k-expression*. Each k-expression t uniquely defines a labeled graph $lab(t)$, where the labels are integers $\{1, \ldots, k\}$ associated with the vertices and each vertex has exactly one label. We say that a k-expression t defines a graph G if G is equal to the graph obtained from the labeled graph $lab(t)$ after removing its labels. The clique-width of a graph G is equal to the minimum k such that there exists a k-expression defining G.

The clique-width of a graph of treewidth k is bounded above by $3 \cdot 2^{k-1}$. This implies that a class of graphs with uniformly bounded treewidth is also of bounded clique-width. The complete graphs show that converse is generally not true. In this sense, showing that a problem can be efficiently solved for graphs of bounded clique-width is more general than showing the same statement for graphs of bounded treewidth.

Many graph problems that are **NP**-hard for general graphs are solvable in linear time when restricted to graphs of clique-width at most k, if a k-expression is given as part of the input.[9] EWIS is no exception.

Theorem 4.19. *For every fixed k,* EWIS *admits an $O(M^2 l)$ algorithm for graphs of clique-width at most k, where l is the number of operations in a given k-expression for G.*

Proof. Suppose that the labels are integers $\{1, \ldots, k\} = [k]$. For every subset of labels $S \subseteq [k]$, let $\mathrm{EWIS}(G, w, S, m)$ denote the answer to the following question: "Is there an independent set of G with total weight m that contains exactly the labels from S?"

Given a k-expression t defining the input graph G, we can solve $\mathrm{EWIS}(G, w, M)$ by first computing all the values for $\mathrm{EWIS}(G, w, S, m)$, for every subset of labels $S \subseteq [k]$, and every $m \in [M]$. It is easy to see that this can be performed in time $O(M^2 l)$ by the following dynamic programming algorithm.

If $|V| = 1$ then let $v \in V$. For all $S \subseteq [k]$, and for all $m \in [M]$, let

$$\mathrm{EWIS}(G, w, S, m) = \begin{cases} yes, & \text{if } S = \{label(v)\} \text{ and } m = w(v); \\ no, & \text{otherwise.} \end{cases}$$

If $G = G_1 \oplus G_2$ then let for all $S \subseteq [k]$, and for all $m \in [M]$:

$$\mathrm{EWIS}(G, w, S, m) = \begin{cases} yes, & \text{if } \mathrm{EWIS}(G_1, w, S, m) = yes; \\ yes, & \text{if } \mathrm{EWIS}(G_2, w, S, m) = yes; \\ yes, & \text{if there is an } m' \in [m-1] \text{ such that} \\ & \quad \mathrm{EWIS}(G_1, w, S, m') = \mathrm{EWIS}(G_2, w, S, m - m') = yes; \\ no, & \text{otherwise.} \end{cases}$$

This can be computed in time $O(M^2)$, similarly as in Corollary 4.2.
If $G = \eta_{i,j}(G_1)$ then let for all $S \subseteq [k]$, and for all $m \in [M]$:

$$\mathrm{EWIS}(G, w, S, m) = \begin{cases} \mathrm{EWIS}(G_1, w, S, m), & \text{if } \{i, j\} \not\subseteq S; \\ no, & \text{otherwise.} \end{cases}$$

If $G = \rho_{i \to j}(G_1)$ then let for all $S \subseteq [k]$, and for all $m \in [M]$:

$$\mathrm{EWIS}(G, w, S, m) = \begin{cases} \mathrm{EWIS}(G_1, w, S, m), & \text{if } S \cap \{i, j\} = \emptyset; \\ \mathrm{EWIS}(G_1, w, S \cup \{i\}, m), & \text{if } S \cap \{i, j\} = \{j\}; \\ no, & \text{otherwise.} \end{cases}$$

[9] If only a graph G of clique-width at most k is given, then an $O(2^{6k})$-expression defining G can be computed in $O(n^3)$ time, as shown by Oum in [41].

Having computed all the values $\mathrm{EWIS}(G, w, S, m)$, the solution to $\mathrm{EWIS}(G, w, M)$ is clearly given by

$$\mathrm{EWIS}(G, w, M) = \begin{cases} yes, & \text{if there is an } S \subseteq [k] \text{ such that } \mathrm{EWIS}(G, w, S, M) = yes; \\ no, & \text{otherwise.} \end{cases}$$

□

Note that the same algorithm runs in pseudo-polynomial time whenever the clique-width of the input graph is of the order $O(\log n)$.

Due to the unknown complexity of the exact perfect matching problem, the problem of determining the complexity of EWIS is of particular interest for line graphs of bipartite graphs, and their subclasses and superclasses. Line graphs of bipartite graphs form a hereditary class of graphs. Their characterization in terms of forbidden induced subgraphs has been obtained in [46], as follows. A graph G is the line graph of a bipartite graph if and only G is $\mathcal{F}$-free, where $\mathcal{F} = \{claw, diamond, C_5, C_7, \ldots\}$, where a *diamond* is the graph obtained by deleting a single edge from a complete graph on 4 vertices.

Keeping in mind this characterization of line graphs of bipartite graphs, it is interesting to consider the following immediate consequence of Theorem 4.19.

Corollary 4.20. EWIS *admits a pseudo-polynomial solution in each of the following graph classes:*

- $(claw, co\text{-}claw)$*-free graphs,*
- $(gem, fork, co\text{-}P)$*-free graphs (see Figure 4) and their subclass* $(claw, diamond, co\text{-}P)$*-free graphs,*
- $(P_5, diamond)$*-free graphs.*

Proof. Each of the above subclasses is of bounded clique-width (see [15, 14, 9]). □

Also, we can derive from Theorem 4.19 a particular complexity result for the exact perfect matching problem.

Corollary 4.21. *For every fixed k, the exact perfect matching problem admits a pseudo-polynomial algorithm for graphs of treewidth at most k.*

Proof. As shown by Gurski and Wanke [26], a set $\mathcal{G}$ of graphs has bounded treewidth if and only if $L(\mathcal{G})$ has bounded clique-width. Since the exact perfect matching problem in $\mathcal{G}$ is polynomially equivalent to the problem EWIS_α in the set $L(\mathcal{G})$, the statement follows from Theorem 4.19 and part (ii) of Lemma 2.1. □

4.2 Modular decomposition

The idea of *modular decomposition* has been first described in the 1960s by Gallai [24], and also appeared in the literature under various other names such as *prime tree decomposition* [22], *X-join decomposition* [27], or *substitution decomposition* [36]. This technique allows one to reduce many graph problems from arbitrary graphs to so-called *prime* graphs. In this subsection, we show how to apply modular decomposition to the EWIS problem.

Let $G = (V, E)$ be a graph, U a subset of V and x a vertex of G outside U. We say that x *distinguishes* U if x has both a neighbor and a non-neighbor in U. A subset $U \subset V(G)$ is called a *module* in G if it is indistinguishable for the vertices outside U. A module U is *nontrivial* if $1 < |U| < |V|$, otherwise it is *trivial*. A graph whose every module is trivial is called *prime*.

An important property of maximal modules is that if G and co-G are both connected, then the maximal modules of G are pairwise disjoint. Moreover, from the above definition it follows that if U and W are maximal modules, then either there are all possible edges between U and W, or there are no edges between them. This property is crucial for the modular decomposition, which provides a reduction of many graph problems from a graph G to the graph G^0 obtained from G by contracting each maximal module to a single vertex.

We formally describe this reduction for the EWIS problem in the recursive procedure MODULAR_EWIS(G, W, M) below. It turns out that in order to apply this decomposition to the EWIS problem, we need to relax the problem so that each vertex of the input graph is equipped with a nonempty set of possible weights (instead of just a single one). For simplicity, we still name this problem EWIS. When all sets are singletons, the problem coincides with the original EWIS problem.

EXACT WEIGHTED INDEPENDENT SET (EWIS)
INSTANCE: An ordered triple (G, W, M), where $G = (V, E)$ is a graph, M is a positive integer and $W = (W_v : v \in V)$ with $W_v \subseteq [M]$ for all $v \in V$ is the collection of possible weights for each vertex of G.
QUESTION: Is there an independent set I of G and a mapping $w : I \to [M]$ such that $w(v) \in W_v$ for all $v \in I$, and $\sum_{v \in I} w(v) = M$?

In graph classes that are closed under duplicating vertices, this extended version is pseudo-polynomially equivalent to the original one: given an input (G, W, M) to the extended version, we can construct a weighted graph (G', w') from (G, W) by replacing each vertex v of G with a clique K_v on $|W_v|$ vertices, assigning different weights from W_v to different vertices of K_v, and joining a vertex from K_u with a vertex from K_v by an edge if and only if $[u, v]$ was an edge of G. Then, it is clear that EWIS$(G, W, M) = yes$ if and only if EWIS$(G', w', M) = yes$. However, working with the extended version enables us to apply modular decomposition to *arbitrary* graph classes.

Let us now describe how modular decomposition can be applied to our problem.[10]

Algorithm MODULAR_EWIS(G, W, M)

Input: An ordered triple (G, W, M), where $G = (V, E)$ is a graph, M is a positive integer and $W = (W_v : v \in V)$ with $W_v \subseteq [M]$ for all $v \in V$ is the collection of possible weights for each vertex of G.

Output: (EWIS$(G, W, k) : k \in [M]$)

1. If $|V| = 1$, say $V = \{v\}$, set, for each $k \in [M]$,

$$\text{EWIS}(G, W, k) = \begin{cases} yes, & \text{if } k \in W_v; \\ no, & \text{otherwise} \end{cases}$$

   and stop.

2. If G is disconnected, partition it into connected components $\mathcal{M}_1, \ldots, \mathcal{M}_r$, and go to step 5.

3. If co-G is disconnected, partition G into co-components $\mathcal{M}_1, \ldots, \mathcal{M}_r$, and go to step 5.

[10] See [32] for a description of the analogous result for the WIS problem.

4. If G and co-G are connected, partition G into maximal modules $\mathcal{M}_1, \ldots, \mathcal{M}_r$.

5. For all $j \in [r]$, let

$$(EWIS(G[\mathcal{M}_j], W, k) : k \in [M]) = \textsc{Modular_ewis}(G[\mathcal{M}_j], W, M)\,.$$

   Construct a graph G^0 from G by contracting each $\mathcal{M}_j$ (for $j \in [r]$) to a single vertex, and assign to that vertex the set of weights

$$W_{\mathcal{M}_j} = \{k \in [M] : \text{EWIS}(G[\mathcal{M}_j], W, k) = yes\}\,.$$

6. For each $k \in [M]$, let

$$\text{EWIS}(G, W, k) = \text{EWIS}(G^0, (W_{\mathcal{M}_j} : j \in [r]), k)$$

   and stop.

We remark that for each input graph, at most one of the steps 2-4 is performed. (At most one among $\{G$, co-$G\}$ is disconnected; moreover, if G and co-G are both connected, then the maximal modules of G are pairwise disjoint.) Observe that the graph G^0 constructed in step 5 of the algorithm is either an edgeless graph, a complete graph, or a prime graph. Therefore, the modular decomposition approach reduces the problem from a graph to its prime induced subgraphs.

The correctness of the procedure is straightforward: every independent set I of G consists of pairwise disjoint independent sets in the subgraphs of G induced by $\mathcal{M}_1, \ldots, \mathcal{M}_r$; moreover, those $\mathcal{M}_i$'s that contain a vertex from I form an independent set in G^0. And conversely, for every independent set I^0 in G^0 and every choice of independent sets $\{I_j : j \in I^0\}$ with I_j independent in $G[\mathcal{M}_j]$, the set $\cup_{j\in[r]} I_j$ is independent in G.

The following theorem answers the question on the complexity of such a reduction.

Theorem 4.22. *Let $\mathcal{G}$ be a class of graphs and $\mathcal{G}^*$ the class of all prime induced subgraphs of the graphs in $\mathcal{G}$. If there is a $p \geq 1$ and a $q \geq 2$ such that* EWIS *can be solved for graphs in $\mathcal{G}^*$ in time $O(M^q n^p)$, then* EWIS *can be solved for graphs in $\mathcal{G}$ in time $O(M^q n^p + m)$.*

Proof. Let G be a graph in $\mathcal{G}$ with n vertices and m edges. The recursive decomposition of G produced by the algorithm can be implemented in time $O(n + m)$ [34]. This decomposition associates with G a tree $T(G)$ whose leaves correspond to the vertices of G, while the internal nodes of $T(G)$ represent induced subgraphs of G with at least two vertices.

Consider an internal node U of $T(G)$, and let G_U denote the induced subgraph of G corresponding to U. Then the children of G_U correspond to the subgraphs $G[\mathcal{M}_1], \ldots, G[\mathcal{M}_r]$, where $\{\mathcal{M}_1, \ldots, \mathcal{M}_r\}$ is the partition of G_U defined in steps 2–4 of the algorithm.

If G_U is disconnected, then G_U^0 is an empty graph, and the problem can be solved for G_U^0 in time $O(M^2|V(G_U^0)|)$, since it is a generalized subset sum problem (cf. Lemma 4.1). If G_U is disconnected, then G_U^0 is a complete graph, and the problem can be solved trivially for G_U^0 in time $O(M|V(G_U^0)|)$.

If both G and co-G are connected, then G_U^0 is a prime induced subgraph of G, and the problem can be solved for G_U^0 in time $O(M^q|V(G_U^0)|^p)$ by our assumption. Summing up over all internal nodes of $T(G)$, we conclude that the total time complexity of the problem on G is bounded by $O(M^q \sum_U |V(G_U^0)|^p)$. It is not difficult to see that the total number of vertices in all graphs G_U^0 corresponding to internal nodes $U \in V(T(G))$ equals to the number of edges of

$T(G)$, i.e., $|V(T(G))| - 1$. Since the number of leaves of $T(G)$ is n and the number of internal nodes is at most $n-1$, we conclude that

$$M^q \sum_U |V(G_U^0)|^p \leq M^q \left(\sum_U |V(G_U^0)| \right)^p \leq M^q (2n-2)^p = O(M^q n^p).$$

Adding the term $O(n+m)$ needed to obtain the decomposition tree, we obtain the desired time complexity. The theorem is proved. □

Just like for the weighted independent set problem, modular decomposition is the key to pseudo-polynomial-time solutions to EWIS in several subclasses of P_5-free and *fork*-free graphs. The results are summarized in the following theorem; all graphs mentioned in the theorem or its proof are depicted in Figure 4.

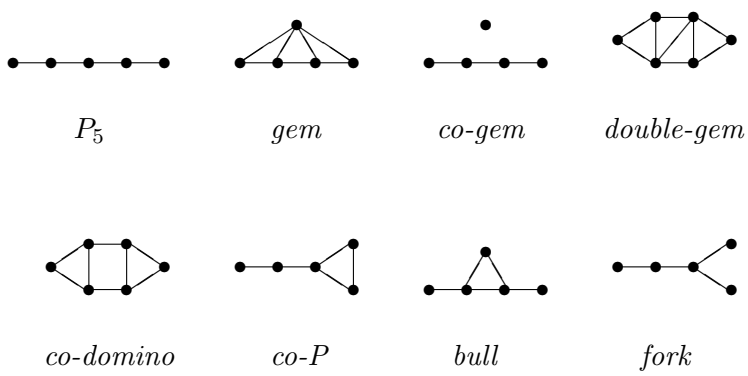

Figure 4: Some 5- and 6-vertex graphs

Theorem 4.23. EWIS *is solvable in pseudo-polynomial time for each of the following classes:*

- (P_5, *double-gem, co-domino*)*-free graphs (and their subclass,* (P_5, *co-P*)*-free graphs),*
- (*bull, fork*)*-free graphs,*
- (*co-P, fork*)*-free graphs,*
- (P_5, *fork*)*-free graphs.*

Proof. This theorem essentially follows from Theorem 4.22 and the results in [12] and [28] (see also [11] for some applications of modular decomposition to the WIS problem). We briefly summarize the main ideas.

Every prime (P_5, *double-gem, co-domino*)-free graph is $2K_2$-free (the complementary version of this statement is proved in [28]). Since we can easily extend Theorem 4.4 to the extended version of EWIS, this implies the result for (P_5, *double-gem, co-domino*)-free graphs.

The (extended) EWIS problem can be solved in pseudo-polynomial time for *co-gem*-free graphs. Indeed, for every vertex v of a *co-gem*-free graph G, the non-neighborhood of v in G is P_4-free. So the problem reduces to solving $O(nM)$ subproblems in P_4-free graphs, which can be done by modular decomposition. It is well known (and can be easily shown) that every P_4-free graph is either disconnected, or its complement is disconnected. Thus, the only prime P_4-free graph is the graph on a single vertex.

In [12], it is shown that prime graphs that contain a *co-gem* and are either $(bull, fork)$-free, $(co\text{-}P, fork)$-free or $(P_5, fork)$-free have a very simple structure. The (extended) EWIS problem can be solved in pseudo-polynomial time for such graphs. Together with the above observation about *co-gem*-free graphs and Theorem 4.22, this concludes the proof. □

5 Concluding remarks

As we saw in the introduction, the motivation for studying the exact weighted independent set problem comes from the fact that the complexity of the exact matching problem is still unknown, even for bipartite graphs. Hence, the problem of determining the complexity of EWIS is of particular interest for line graphs of bipartite graphs, and their subclasses and superclasses. We will now show that the class $L(Bip)$ of line graphs of bipartite graphs is sandwiched between two graph classes for which the complexity of EWIS is known, and whose (infinite) sets of forbidden induced subgraphs differ only in two graphs.

Recall that the line graphs of bipartite graphs are precisely the $(claw, diamond, C_5, C_7, \ldots)$-free graphs. Replacing the diamond in the above characterization by its subgraph C_3 results in a smaller class of $(claw, C_3, C_5, C_7, \ldots)$-free graphs. It is easy to see that this is precisely the class of bipartite graphs of maximum degree 2. Every connected graph in this class is either an even cycle or a path, and the treewidth of such graphs is at most 2. By Corollary 4.2 and Theorem 4.18, the problem is solvable in pseudo-polynomial time in this class.

On the other hand, if we replace the $claw = K_{1,3}$ with $K_{1,4}$ in the above characterization of $L(Bip)$, we obtain a class of graphs that strictly contains line graphs of bipartite graphs. This class of $(K_{1,4}, diamond, C_5, C_7, \ldots)$-free graphs contains the class of $(K_{1,4}, C_3, C_5, C_7, \ldots)$-free graphs, which is precisely the class of bipartite graphs of maximum degree at most 3. The results of Section 3.1 imply that the problem is strongly **NP**-complete for this class, and hence also for the larger class of $(K_{1,4}, diamond, C_5, C_7, \ldots)$-free graphs.

To summarize, the class $L(Bip)$ of line graphs of bipartite graphs is sandwiched between two graph classes for which the complexity of EWIS is known, as the following diagram shows.

$Free(\{claw, C_3, C_5, C_7, \ldots\})$	$\subset$	$L(Bip)$	$\subset$	$Free(\{K_{1,4}, diamond, C_5, C_7, \ldots\})$
pseudo-polynomial		???		strongly NP-complete

References

[1] V.E. Alekseev, "On the number of maximal independent sets in graphs from hereditary classes," Combinatorial-algebraic methods in discrete optimization, University of Nizhny Novgorod, 1991, pp. 58 (in Russian).

[2] S. Arnborg and A. Proskurowski, "Linear time algorithms for NP-hard problems restricted to partial k-trees," *Discrete Appl. Math.* **23** (1989), 11–24.

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

[4] H.-J. Bandelt and H.M. Mulder, "Distance-hereditary graphs," *J. Combin. Theory Ser. B* **41** (1986) 182–208.

[5] F. Barahona and W.R. Pulleyblank, "Exact arborescences, matchings, and cycles," *Discrete Appl. Math.* **16** (1987) 91-99.

[6] C. Berge, *Graphs and hypergraphs.* North Holland, Amsterdam, 1973.

[7] J. Błażewicz, P. Formanowicz, M. Kasprzak, P. Schuurman and G. Woeginger, "A polynomial time equivalence between DNA sequencing and the exact perfect matching problem," *Discrete Optim.* (2006), doi:10.1016/j.disopt.2006.07.004

[8] H.L. Bodlaender and T. Kloks, "Efficient and constructive algorithms for the pathwidth and treewidth of graphs," *J. Algorithms* **21** (1996) 358–402.

[9] A. Brandstädt, "(P_5, diamond)-free graphs revisited: structure and linear time optimization," *Discrete Appl. Math.* **138** (2004) 13–27.

[10] A. Brandstädt and F.F. Dragan, "On linear and circular structure of (claw, net)-free graphs," *Discrete Appl. Math.* **129** (2003) 285–303.

[11] A. Brandstädt, C.T. Hoàng and J.-M. Vanherpe, "On minimal prime extensions of a four-vertex graph in a prime graph," *Discrete Math.* **288** (2004) 9–17.

[12] A. Brandstädt, V.B. Le and H.N. de Ridder, "Efficient robust algorithms for the maximum weight stable set problem in chair-free graph classes," *Inform. Process. Lett.* **89** (2004) 165-173.

[13] A. Brandstädt, V.B. Le and J. Spinrad, *Graph classes: a survey.* SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 1999.

[14] A. Brandstädt, H.-O. Le and J.-M. Vanherpe, "Structure and stability number of chair-, co-P- and gem-free graphs revisited," *Inform. Process. Lett.* **86** (2003) 161-167.

[15] A. Brandstädt and S. Mahfud, "Maximum weight stable set on graphs without claw and co-claw (and similar graph classes) can be solved in linear time," *Inform. Process. Lett.* **84** (2002) 251–259.

[16] H. Broersma, T. Kloks, D. Kratsch and H. Müller, "Independent sets in asteroidal triple-free graphs," *SIAM J. Discrete Math.* **12** (1999) 276–287.

[17] O. Cogis and E. Thierry, "Computing maximum stable sets for distance-hereditary graphs," *Discrete Optim.* **2** (2005) 185–188.

[18] D.G. Corneil and U. Rotics, "On the relationship between clique-width and treewidth," *SIAM J. Comput.* **34** (2005) 825–847.

[19] G. Damiand, M. Habib and C. Paul, "A simple paradigm for graph recognition: application to cographs and distance hereditary graphs." Combinatorics and computer science (Palaiseau, 1997). *Theoret. Comput. Sci.* **263** (2001) 99–111.

[20] M. Dawande, P. Keskinocak, J.M. Swaminathan and S. Tayur, "On Bipartite and Multipartite Clique Problems," *J. Algorithms* **41** (2001) 388–403.

[21] V.G. Deĭneko and G.J. Woeginger, "On the robust assignment problem under a fixed number of cost scenarios," *Oper. Res. Lett.* **34** (2006) 175–179.

[22] A. Ehrenfeucht and G. Rozenberg, "Primitivity is hereditary for 2-structures," *Theoret. Comput. Sci.* 70 (1990) 343–358.

[23] A. Frank, "Some polynomial algorithms for certain graphs and hypergraphs," *Proc. of the 5th Brit. Comb. Conf., Aberdeen 1975*, Congr. Numer. XV, 211-226 (1976).

[24] T. Gallai, "Transitiv orientierbare graphen," *Acta Math. Acad. Sci. Hungar.* **18** (1967) 25–66.

[25] M.R. Garey and D.S. Johnson (1979). Computers and intractability. A guide to the theory of NP-completeness. *CA, Freeman.*

[26] F. Gurski and E. Wanke, "Line graphs of bounded clique-width," *Discrete Appl. Math.* to appear, 2007.

[27] M. Habib and M.C. Maurer, "On the X-join decomposition for undirected graphs," *Discrete Appl. Math.* 1 (1979) 201–207.

[28] C.T. Hoáng and B. Reed, "Some classes of perfectly orderable graphs," *J. Graph Theory* **13** (1989) 445-463.

[29] W.-L. Hsu and T.-H. Ma, "Fast and simple algorithms for recognizing chordal comparability graphs and interval graphs," *SIAM J. Comput.* **28** (1999), 1004-1020.

[30] A.V. Karzanov, "Maximum matching of given weight in complete and complete bipartite graphs," *Cybernetics* **23** (1987) 8-13; translation from *Kibernetika* **1** (1987) 7-11.

[31] M. Leclerc, "Polynomial time algorithms for exact matching problems," Masters Thesis, University of Waterloo, Waterloo, 1986.

[32] V.V. Lozin and M. Milanič, "A polynomial algorithm to find an independent set of maximum weight in a fork-free graph," *Proc. of the ACM-SIAM Sympos. on Discrete Algorithms SODA'06* (2006) 26–30.

[33] C. Mannino, G. Oriolo, F. Ricci and S. Chandran. "The stable set problem and the thinness of a graph," *Oper. Res. Lett.* **35** (2007) 1–9.

[34] R.M. McConnell and J. P. Spinrad, "Modular decomposition and transitive orientation," *Discrete Math.* **201** (1999) 199–241.

[35] M. Milanič and J. Monnot, "On the complexity of the exact weighted independent set problem for various graph classses," DIMACS Technical Report 2006-17, available online at `ftp://dimacs.rutgers.edu/pub/dimacs/TechnicalReports/TechReports/2006/2006-17.ps.gz`

[36] R.H. Möhring, "Algorithmic aspects of comparability graphs and interval graphs," in: I. Rival (Ed.), *Graphs and Orders*, D. Reidel, Boston, 1985, 41–101.

[37] R.H. Möhring and F.J. Radermacher, Substitution decomposition for discrete structures and connections with combinatorial optimization. Algebraic and combinatorial methods in operations research, 257–355, North-Holland Math. Stud., 95, North-Holland, Amterdam, 1984.

[38] K. Mulmuley, U. Vazirani and V.V. Vazirani, "Matching is as easy as matrix inversion," *Combinatorica* **7** (1987) 105-113.

[39] O.J. Murphy, "Computing independent sets in graphs with large girth," *Discrete Appl. Math.* **35** (1992) 167–170.

[40] Y. Okamoto, T. Uno and R. Uehara, "Linear-time counting algorithms for independent sets in chordal graphs," Graph-theoretic concepts in computer science, 433–444, *Lecture Notes in Comput. Sci.* **3787**, Springer, Berlin, 2005.

[41] S.-i. Oum, "Approximating rank-width and clique-width quickly," Graph-theoretic concepts in computer science, 49–58, *Lecture Notes in Comput. Sci.* **3787**, Springer, Berlin, 2005.

[42] C.H. Papadimitriou and M. Yannakakis, "The complexity of restricted spanning tree problems," *J. ACM* **29** (1982) 285-309.

[43] S. Poljak, "A note on stable sets and coloring of graphs," *Comment. Math. Univ. Carolinae* **15** (1974) 307–309.

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

[45] J. Ramalingam and C. Pandu Rangan, "A unified approach to domination problems on interval graphs," *Inform. Process. Lett.* **27** (1988) 271-274.

[46] W. Staton and G.C. Wingard, "On line graphs of bipartite graphs," *Util. Math.* **53** (1998) 183–187.

[47] K.J. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Trans. on CAD of ICAS* **CAD-6** (1987) 93–94.

[48] R.E. Tarjan, "Decomposition by clique separators," *Discrete Math.* **55** (1985) 221–232.

[49] R. Thomas, "A survey of Pfaffian orientations of graphs," In *Proceedings of the International Congress of Mathematicians, Vol. III (Madrid, 2006),* 963–984, EMS Publishing House, 2006.

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