DIMACS Technical Report 2006-17
September 2006

On the complexity of the exact weighted independent set problem for various graph classes

by

Martin Milanič[1]
RUTCOR, Rutgers University
640 Bartholomew Road, Piscataway
NJ 08854-8003, USA
E-mail: mmilanic@rutcor.rutgers.edu

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

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

ABSTRACT

The *exact weighted independent set* problem (EWIS) consists of determining whether a given weighted graph contains an independent set whose weight equals a given integer M. We study the problem of determining the complexity of the exact weighted independent set problem, and its restricted version EWIS_α (where the independent set is required to be of maximum size), for particular graph classes. We prove that 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 distinguish several graph classes where EWIS and EWIS_α can be solved in pseudo-polynomial time.

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

1 Introduction

An *independent set* of 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 the vertex weights are equal to 1, we speak about the *independent set* problem (IS). The optimal values of these problems are denoted by $\alpha_w(G)$ and $\alpha(G)$, respectively.

A related problem and one of the main interest in this paper is the *exact weighted independent set* problem (EWIS), which 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. Formally, the solution to $\text{EWIS}(G, w, M)$ is YES if and only if there is an independent set I of G with $w(I) = M$. The restriction where we require the independent set to be of maximum size will be denoted by EWIS_α. Thus, $\text{EWIS}_\alpha(G, w, M)$ asks for the existence of an independent set I of G with $|I| = \alpha(G)$ and weight M.

In this paper, we focus on the problem of determining the complexity of EWIS and EWIS_α for particular graph classes. On one hand, we present the first nontrivial strong **NP**-completeness result for EWIS. On the other hand, we distinguish several graph classes where EWIS can be solved in pseudo-polynomial time.

Exact Problems. 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 perfect graph. How hard is it to determine whether there exists a solution with a given weight? Papadimitriou and Yannakakis showed in [20] 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. Known results can be summarized as follows.

- The *exact spanning tree* problem, and more generally, the *exact arborescence* problem are solvable in pseudo-polynomial time [2].

- The *exact cut* problem is solvable in pseudo-polynomial time for planar and toroidal graphs [2].

- The *exact perfect matching* problem is solvable in pseudo-polynomial time for planar graphs [2], and more generally, for graphs that have a Pfaffian orientation (provided one is given). Bipartite graphs with a Pfaffian orientation have been characterized in [24], where a polynomial-time recognition algorithm is also presented. Karzanov [14] 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 [20] that the problem for general (or bipartite) graphs with weights encoded in unary is polynomially reducible to the one with 0-1 weights. Mulmuley *et al.* [18] 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.

The exact perfect matching problem is of great practical importance. It has applications in diverse areas such as bus-driver scheduling and statistical mechanics (see [15]), DNA sequencing [4] and robust assignment problems [10]. The fact that its complexity is still unknown motivates us to study the exact weighted independent set problem (EWIS) and its restricted version EWIS_α, both closely related to the exact perfect matching problem. There is a one-to-one correspondence between the matchings of a graph G and the independent sets of the line graph of G.[1] 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 problem EWIS_α, restricted to the class of the line graphs of graphs with a perfect matching.

Our results. We first show some relations between the complexities of EWIS and EWIS_α, when restricted to particular graph classes. The main results of this article, however, can be summarized in the spirit of the above list of complexity results on exact problems as follows:

- The *exact weighted independent set* problem and its restricted version EWIS_α are strongly **NP**-complete for cubic bipartite graphs. (We actually extend this result to a much more general one, but since we would need a couple of definitions to state it, we postpone the detailed formulation until Section 3.2.)

- The *exact weighted independent set* problem and its restricted version EWIS_α are solvable in pseudo-polynomial time for any of the following graph classes: mK_2-free graphs, AT-free graphs, k-thin graphs, circle graphs, and graphs of bounded clique-width.

Note that in view of the relation between the exact perfect matching problem and the problem EWIS_α, each of these polynomial results also gives a polynomial result for the exact perfect matching problem.

Notation and organization. All graphs considered are simple, without loops and undirected. Unless otherwise stated, n and m will denote the number of vertices and edges, respectively, of the graph considered. For graph theoretical terms not defined here, the reader is referred to Berge's book [3].

The triple (G, w, M), with $G = (V, E)$, will always represent an instance of EWIS (or EWIS_α). 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 $V(H)$. Finally, for a subset of vertices $V' \subseteq V$, we let $w(V') = \sum_{v\in V'} w(v)$.

[1] 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.

The remainder of the paper is organized as follows. In Section 2, we continue the introductory discussion and present some polynomial preprocessing steps that we will later on assume performed. We also discuss some 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 based on dynamic programming 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 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 [13]). 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 $J \subseteq \{1, \ldots, n\}$ 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 $v \in V(G)$. 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 $k \in \{1, \ldots, n\}$ (assuming M is nonzero, 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 $k \in \{1, \ldots, n\}$, 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.

The same assumption on vertex weights as for EWIS can be made for the instances (G, w, M) of 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 $k = \alpha(G)$, which is equal to that only $p \in \{1, \ldots, n\}$ 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 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_α *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_α *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_α 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_α problem. It is easy to see that the following algorithm solves EWIS_α.
Step 1. Compute $\alpha(G)$, which is equal to the maximum $k \in \{1, \ldots, n\}$ such that the solution to $\mathrm{EWIS}(G, \mathbf{1}, k)$ 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 $\mathrm{EWIS}_\alpha(G, w, M) = \mathrm{EWIS}(G, w', M')$.
(iii) Let (G, w, M) with $G = (V, E) \in \mathcal{G}$ be an instance of the exact weighted independent set problem. Let G' be the graph, defined as in the lemma. Let $n = |V(G)|$ 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 $\mathrm{EWIS}(G, w, M)$ is YES if and only if the solution to $\mathrm{EWIS}_\alpha(G', w', M')$ is YES for some value M' in the set $\{(n+1)M, \ldots, (n+1)M + n - 1\}$. □

The problem EWIS is clearly in **NP**, and so is EWIS_α for any class of graphs $\mathcal{G}$ where IS is polynomially solvable. Therefore, Lemma 1 implies the following result.

Corollary 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_α *is strongly **NP**-hard for graphs in $\mathcal{G}$. If, in addition,* IS *is polynomial for graphs in $\mathcal{G}$, then* EWIS_α *is strongly **NP**-complete for graphs in $\mathcal{G}$.*
(ii) If EWIS_α *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 1. If EWIS *is strongly **NP**-complete for graphs in $\mathcal{G}$, then* EWIS_α *is strongly **NP**-hard for graphs in $\mathcal{G}'$. If, in addition,* IS *is polynomial for graphs in $\mathcal{G}'$, then* EWIS_α *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_α 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).

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 class of line graphs, line graphs of bipartite graphs, line graph of complete graphs with an even number of vertices, and line graph of complete balanced bipartite graphs, respectively, we have the following result.

Lemma 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 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 $k \in \{0, \dots, n-1\}$. 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

3.1 Bipartite graphs

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 [13, 9]. This problem consists in deciding, given a bipartite graph $G = (L, R; E)$ and an integer k, if there exists $L' \subseteq L$ 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 [9], 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 add a weight 1 on vertices of L and $B = \max\{|L|, |R|\} + 1$ on vertices 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$_\alpha$ is strongly **NP**-complete even for cubic bipartite graphs. We can observe that for graphs of maximum degree 2, EWIS and EWIS$_\alpha$ are pseudo-polynomially solvable problems. Every connected graph in this class is either a cycle or a path, and the treewidth of such graphs is at most 2.[2] By Corollary 7 and Theorem 20, the problem is solvable in pseudo-polynomial time in this class.

[2]For the definition of treewidth, see for instance [11].

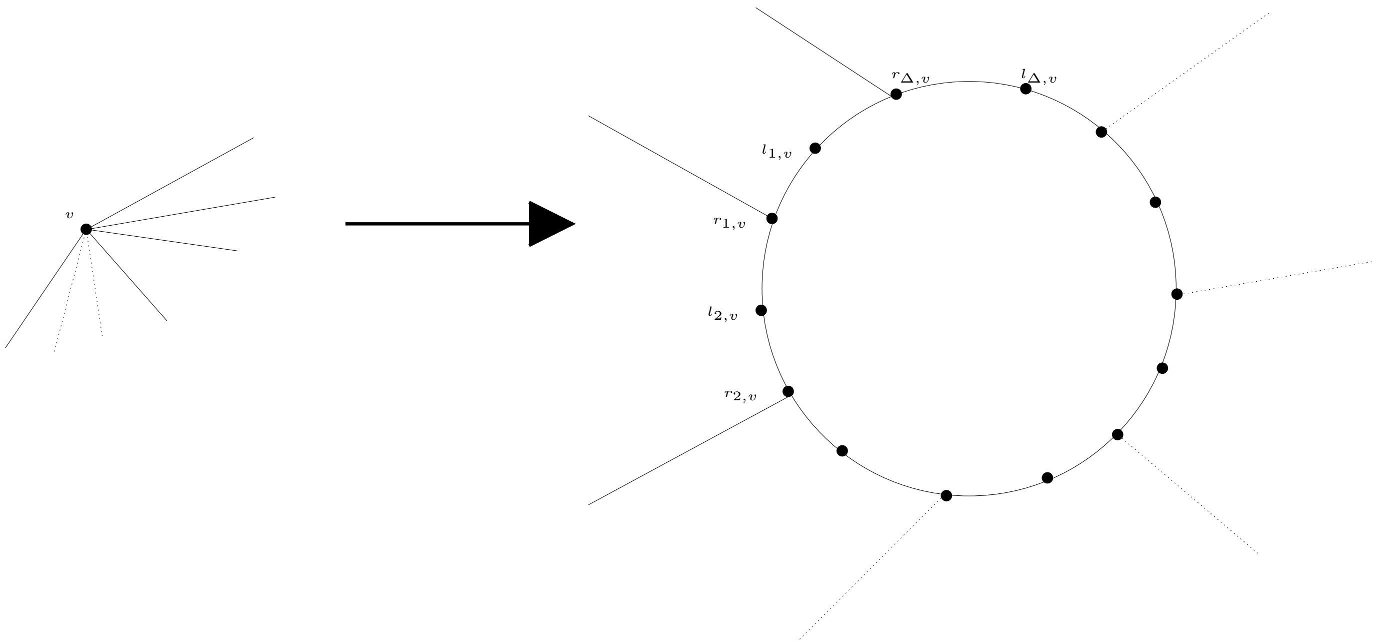

Figure 1: The gadget $H(v)$.

Theorem 4. EWIS$_\alpha$ *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 reduction is made from the decision clique set problem in regular graph which is known to be **NP**-complete, [13]. 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. W.l.o.g., assume that $0 < k < \Delta < n - 1$, since otherwise the problem is easy. We build the instance $I = (G', w)$ of EWIS where $G' = (L, R; E')$ is a bipartite graph as follows:

- 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 $i = 1, \ldots, \Delta$. 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 applying iteratively 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

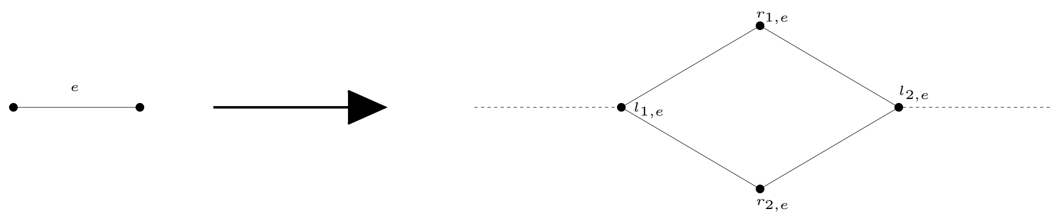

Figure 2: The gadget $H(e)$.

$l_{j,v}$ 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 a degree 3.

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

We claim that there exist a clique V^* of G with a size at least k if and only if $\text{EWIS}(G', w, M)$ 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 for each gadget $H(v)$ and $H(e)$ a perfect matching), we deduce 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 $l_{i,e} \in I$ iff $L_e \subseteq I$, 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:

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

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 V^* is a clique of G.

Conversely, let V' be a clique of G with $\mid V' \mid \geq k$ 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 verifies that $w(I) = M$ and that I is a maximum independent set of G'. Indeed, let us assume the reverse; 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 now complete. □

As corollary of Theorem 4, 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 4 by a cycle of length $2n\Delta$ and we delete edges $[l_{i,u}, r_{1,e}]$ and $[l_{j,v}, r_{2,e}]$.

3.2 A more general hardness result

Let us now strengthen the main result of the previous subsection. To this end, we first introduce some notations. We will denote the class of graphs containing no induced subgraphs from a set F by $Free(F)$. Any graph in $Free(F)$ will be called F-free. Our hardness results will be expressed in terms of restrictions on the set of forbidden induced subgraphs F.

By $c(G)$ we denote the *chordality* of G, i.e., the size of a largest chordless cycle in G. (If G contains no cycles, $c(G)$ is defined to be 0.) For a nonempty set of graphs F, we write $c(F) := \max\{c(G) : G \in F\}$. By $h(G)$ we denote the largest i such that G contains an induced subgraph isomorphic to the graph H_i represented on the right in Figure 3. (If G contains no induced graphs of the form H_i, we define $h(G)$ to be 0.) For a nonempty set of graphs F, we write $h(F) := \max\{h(G) : G \in F\}$. Also, we denote by $S_{i,j,k}$ the graph represented on the left in Figure 3, and by $\mathcal{S}$ the class of graphs every connected component of which is of the form $S_{i,j,k}$ for some values of indices i, j and k (depending on the component).

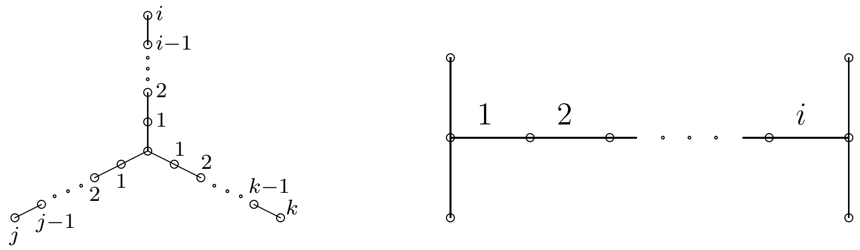

Figure 3: Graphs $S_{i,j,k}$ (left) and H_i (right)

With these definitions in mind and following the idea of Lozin as in Theorem 1 of [16], we can now show the following hardness result.

Theorem 5. *Let $\mathcal{G}$ be a class of F-free bipartite graphs of maximum degree at most 3, such that the set F satisfies the following three conditions:*

(1) $c(F) < \infty$*,*

(2) $h(F) < \infty$*,*

(3) $F \cap \mathcal{S} = \emptyset$*.*

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, let us first that for any fixed k, the problem is strongly **NP**-complete in the class of $(C_3, \ldots, C_k, H_1, \ldots, H_k)$-free bipartite graphs of maximum degree at most 3. So let (G, w, M) be an instance of EWIS_α where G is a bipartite graphs 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.

Note that G' is also a bipartite graph of maximum degree at most 3. Moreover, if G' contains a cycle C, then the length of C is at least $3 \cdot (2k'+1) \geq k+1$. Similarly, if G' contains an induced copy of H_i, then i is at least $k+1$. Therefore, G' is a $(C_3, \ldots, C_k, H_1, \ldots, H_k)$-free bipartite graph of maximum degree at most 3.

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 we recall that $m = |E(G)|$.

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.

We have just showed that EWIS_α is strongly **NP**-complete in the class of $(C_3, \ldots, C_k, H_1, \ldots, H_k)$-free bipartite graphs of maximum degree at most 3.

To prove strong **NP**-completeness of the problem in the class $\mathcal{G}$, we will show that there is a k such that $\mathcal{G}$ contains all $(C_3, \ldots, C_k, H_1, \ldots, H_k)$-free bipartite graphs of maximum degree at most 3. Denote $k := \max(c(F), h(F))$ and let G be a $(C_3, \ldots, C_k, H_1, \ldots, H_k)$-free bipartite graph of maximum degree at most 3.

Assume that G does not belong to $\mathcal{G}$. Then G contains a graph $A \in F$ as an induced subgraph. From the choice of G we know that A has no vertices of degree greater than 3.

For the same reason, A is $(C_3, \ldots, C_k)$-free. Moreover, A cannot contain any cycle of length greater than k, since $F \subset Free(C_{k+1}, C_{k+2}, \ldots)$. Therefore, A is a forest. Similarly, A contains no graph of the form H_i. But then A belongs to $\mathcal{S}$, which contradicts the initial assumption. Hence, $G \in \mathcal{G}$ and the theorem is proved. □

4 Dynamic programming solutions to EWIS

In this section, we present pseudo-polynomial time solutions to the EWIS problem for the following graph classes:

- mK_2-free graphs,
- interval graphs, and two graph classes strictly containing interval graphs: AT-free graphs and k-thin graphs,
- distance-hereditary graphs, and their generalization, circle graphs,
- graphs of bounded treewidth, and their generalization, graphs of bounded clique-width.

All the algorithms are based on dynamic programming. Before stating the algorithms, we present a preliminary result that will be frequently used in our solutions.

The SUBSET SUM problem has been already mentioned in the introduction, and is a typical example of an **NP**-complete problem that can be solved in pseudo-polynomial time by dynamic programming. For the sake of completeness, we recall the problem here.

SUBSET SUM
INSTANCE: Positive integers $a_1, \ldots, a_n, b$.
QUESTION: Is there a subset S of $\{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = b$?

For our algorithmic purposes, we will consider the following generalization of SUBSET SUM.

GENERALIZED SUBSET SUM
INSTANCE: Nonempty sets of positive integers $A_1, \ldots, A_n$ and a positive integer b.
QUESTION: Is there a nonempty subset S of $\{1, \ldots, n\}$ 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 6. 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 $b' \in \{1, \ldots, b\}$ with such property. Clearly, such a set can have size at most b. Having constructed the set B, we can solve GENERALIZED SUBSET SUM by verifying whether $b \in B$, which can be done in time $O(b)$.

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 GENERALIZED SUBSET SUM problem. Let B' be the inductively constructed set of all possible values of $b' \in \{1, \ldots, b\}$ such that the solution to the GENERALIZED SUBSET SUM 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 $\beta \in \{1, \ldots, b\}$. Then, β will belong to B, i.e., the solution to the GENERALIZED SUBSET SUM, 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. (In fact, we could have written $O(\max\{b, (n-1)b^2\})$ instead of $O(nb^2)$ in the statement of the lemma, but we will prefer to use the notationally simpler and asymptotically equivalent time complexity $O(nb^2)$.) □

This result can be efficiently exploited while developing dynamic programming algorithms for the EWIS problem. Namely, it shows that it suffices to consider connected graphs.

Corollary 7. *Let (G, w, M) be an instance of* EWIS*, where $G = (V, E)$ is a graph whose connected components are $C_1, \ldots, C_r$.*

Suppose that for each $i \in \{1, \ldots, r\}$ and for each $k \in \{1, \ldots, M\}$, the solution (either YES *or* NO*) to EWIS(C_i, w, k) is given. Then, we can compute the set of solutions to EWIS(G, w, k) for all $k \in \{0, 1, \ldots, M\}$ in time $O(rM^2)$.*

Proof. Clearly, the solution to EWIS$(G, w, 0)$ is always YES, as the empty set is independent. For $k \in \{1, \ldots, M\}$, it is enough to observe that for every such k, the solution to EWIS(G, w, k) is YES if and only if the solution to the GENERALIZED SUBSET SUM problem on the instance $(A_1, \ldots, A_r; k)$ is YES, where A_i denotes the set of all values $k' \in \{1, \ldots, M\}$ such that the solution to EWIS(C_i, w, k') is YES. □

In the remaining subsections, we present dynamic programming solutions to EWIS for the above-mentioned graph classes.

4.1 mK_2-free graphs

Theorem 8. *For every positive integer m,* EWIS *admits a pseudo-polynomial algorithm for mK_2-free graphs.*

Proof. An mK_2-free graph contains only polynomially many maximal independent sets (see [12]). Tsukiyama *et al.* describe in [25] an algorithm that generates all the maximal independent sets of a graph with polynomial delay. This implies that we can enumerate all maximal independent sets $I_1, \ldots, I_N$ of a given mK_2-free graph G in polynomial time.

For each vertex x of G, the solution to EWIS$(G[\{x\}], p, k)$ is YES if and only if $k \in \{0, w(x)\}$. Therefore, we can apply Corollary 7 to each maximal independent set I of G in order to compute the set of solutions to EWIS$(G[I], p, k)$ for all $k \in \{0, 1, \ldots, M\}$ in pseudo-polynomial time.

As each independent set of G is contained in some maximal independent set I_i of G, the solution to EWIS(G, w, p) is YES if and only if there is an $i \in \{1, \ldots, N\}$ such that the solution to EWIS(I_i, w, p) is YES. This shows that the EWIS problem can be solved in pseudo-polynomial time for mK_2-free graphs. □

4.2 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.

A representation of interval graphs that is particularly suitable for the EWIS problem is the following. It has been shown in [21] 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 9. 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 $i \in \{1, \ldots, n\}$, let G_i denote the subgraph of G induced by $\{v_1, \ldots, v_i\}$. Then, for every $i \in \{2, \ldots, n\}$, 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. For all $i \in \{1, \ldots, n\}$ and all $k \in \{0, 1, \ldots, M\}$, let the solution to EWIS$(G[\{v_i\}], w, k)$ be YES if $k \in \{0, w(v_i)\}$, and NO otherwise.
Step 3. For $i = 2, \ldots, n$, do the following:

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

3.2. For every $k \in \{0, 1, \dots, M\}$, let the solution to EWIS(G_i, w, k) be 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 NO otherwise.

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

4.3 k-thin graphs

The property used in the above characterization of interval graphs has been generalized by Mannino *et al.* in [17], 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 $(v_1, \dots, v_n)$ of V and a partition of V into k classes such that, for each triple (r, s, t) with $1 \le r < s < t \le n$, 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 [17].

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

Theorem 10. *Suppose that for a k-thin graph $G = (V, E)$, $k \ge 2$, an ordering $(v_1, \dots, v_n)$ of V and a partition of V into k classes are given such that, for each triple (r, s, t) with $1 \le r < s < t \le n$, 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 9. Let $V_1, \dots, V_k$ be the classes of the partition. Instead of the graphs G_i, induced by the first i vertices, we now consider all graphs $G(i_1, \dots, i_k)$, induced by the "first" i_r vertices of each class (according to the ordering on V restricted to the class), for all $r \in \{1, \dots, k\}$, and for all $O(n^k)$ choices of such k-tuples $(i_1, \dots, i_k) \in \{1, \dots, |V_1|\} \times \dots \times \{1, \dots, |V_k|\}$. □

4.4 AT-free graphs

Another generalization of interval graphs is given by the so-called *AT-free graphs*. Besides interval graphs, the family of AT-free graphs contains 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 vertices in a graph G is an *AT* (*asteroidal triple*) if for every two of these vertices there is a path between them avoiding the closed neighborhood of the third. A graph is called *AT-free* if it has no AT.

Our dynamic programming algorithm that solves EWIS for AT-free graphs is based on the dynamic programming solution to the WIS problem in AT-free graphs, developed by Broersma, Kloks, Kratsch and Müller in [5].

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 [5].

Theorem 11 (Broersma, Kloks, Kratsch and Müller [5])**.** *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 12 (Broersma, Kloks, Kratsch and Müller [5])**.** *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 13 (Broersma, Kloks, Kratsch and Müller [5])**.** *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 $I(x, y) = \cup_{i=1}^{l} B_i$.*

The following general lemma is obvious.

Lemma 14. *Let (G, w) be a weighted graph, $G = (V, E)$. Then, the solution to EWIS(G, k) is* YES *if and only if there is a vertex $x \in V$ such that the solution to EWIS$(G - N(x), k)$ is* YES*.*

Combining Lemma 14 with Theorems 12 and 13, we obtain the following lemma.

Lemma 15. *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 EWIS(C, w, k) is* YES *if and only if there is a vertex $y \in C$ such that the solution to 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.*

Combining Lemma 14 with Theorem 11, we obtain the following lemma.

Lemma 16. *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 EWIS(I, w, k) is* YES *if and only if there is a vertex $s \in I$ such that the solution to 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.*

Theorem 17. 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 15 and Corollary 7. 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 16 and Corollary 7.
Step 5. Compute the solution to EWIS(G, w, k) using Lemma 14 and Corollary 7. $\square$

4.5 Distance-hereditary graphs

Distance-hereditary graphs are graphs such that the distance between any two connected vertices is the same in every induced subgraph in which they remain connected. Bandelt and Mulder provided in [1] 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.

More formally, a *pruning sequence* of a distance-hereditary graph G is a sequence of the form $\sigma = (x_1 R_1 y_1, x_2 R_2 y_2, \ldots, x_{n-1} R_{n-1} y_{n-1})$ where $(x_1, \ldots, x_n)$ is a total ordering of $V(G)$ such that for all $i \in \{1, \ldots, n-1\}$, the following holds:

- $R_i \in \{P, T, F\}$.
- If we denote by G_i the subgraph of G induced by $\{x_i, \ldots, x_n\}$, then:
  - If $R_i = P$ then x_i is a *pendant vertex*, that is, a vertex of degree one in the graph G_i, with $N_{G_i}(x_i) = \{y_i\}$.
  - If $R_i = T$ then x_i and y_i are *true twins* in G_i, that is, $N_{G_i}[x_i] = N_{G_i}[y_i]$.
  - If $R_i = F$ then x_i and y_i are *false twins* in G_i, that is, $N_{G_i}(x_i) = N_{G_i}(y_i)$.

A pruning sequence of a distance-hereditary graph can be computed in linear time [8]. Our algorithm for EWIS on distance-hereditary graphs is based on the dynamic programming solution for the WIS problem, developed by Cogis and Thierry in [6].

Theorem 18. EWIS *admits an $O(M^2 n + m)$ algorithm for distance-hereditary graphs.*

Proof. Before we describe the algorithm, let us introduce some notation. For two 0-1 functions p and q, defined on $\{0,1,\ldots,N\}$, we denote by $p+q$ the 0-1 function, defined on $\{0,1,\ldots,N\}$, given by the following rule: for every $k \in \{0,1,\ldots,N\}$, we have

$$(p+q)(k) = \begin{cases} 1, & \text{if there is a } k' \in \{0,1,\ldots,k\} \text{ such that } p(k') = q(k-k') = 1; \\ 0, & \text{otherwise.} \end{cases}$$

With this definition in mind, we can describe the algorithm for solving EWIS(G,w,M), where G is a distance-hereditary graph, as follows:

Step 1. To each vertex $x \in V(G)$, associate a pair of functions $p^x : \{0,1,\ldots,M\} \to \{0,1\}$ and $q^x : \{0,1,\ldots,M\} \to \{0,1\}$. For each $x \in V(G)$ and each $k \in \{0,1,\ldots,M\}$, let

$$p^x(k) = \begin{cases} 1, & \text{if } k = w(x); \\ 0, & \text{otherwise,} \end{cases}$$

and

$$q^x(k) = \begin{cases} 1, & \text{if } k = 0; \\ 0, & \text{otherwise.} \end{cases}$$

Step 2. Compute the pruning sequence $\sigma = (x_1R_1y_1, x_2R_2y_2, \ldots, x_{n-1}R_{n-1}y_{n-1})$ for G.
Step 3. Check if the pruning sequence is empty. If yes, there is only one vertex x left. If $\max(p^x(M), q^x(M)) = 1$, then output YES. Else, output NO.
Else, let xRy be the head of the pruning sequence. Update the pruning sequence by removing xRy from it. Update p^y and q^y as follows.

- If $R = P$ then let

$$\begin{aligned} p^y(k) &\leftarrow (p^y + q^x)(k)\,, \\ q^y(k) &\leftarrow \max((p^x + q^y)(k), (q^x + q^y)(k))\,, \end{aligned}$$

  for each $k \in \{0,1,\ldots,M\}$.

- If $R = T$ then let

$$\begin{aligned} p^y(k) &\leftarrow \max((p^y + q^x)(k), (p^x + q^y)(k))\,, \\ q^y(k) &\leftarrow (q^x + q^y)(k)\,, \end{aligned}$$

  for each $k \in \{0,1,\ldots,M\}$.

- If $R = F$ then let

$$\begin{aligned} p^y(k) &\leftarrow \max((p^x + q^y)(k), (p^x + p^y)(k), (q^x + p^y)(k))\,, \\ q^y(k) &\leftarrow (q^x + q^y)(k)\,, \end{aligned}$$

  for each $k \in \{0,1,\ldots,M\}$.

Step 4. Go to Step 3.

To prove the correctness of the algorithm, we first define the following auxiliary problem:

P1(G, M, p, q)
INSTANCE: A graph G, a positive integer M, and two 0-1 functions p and q defined on $V \times \{0, 1, \dots, M\}$.
INSTANCE: Is there an independent set I of G, and a mapping $w : V \to \{0, 1, \dots, M\}$ such that the following holds:

- $\sum_{x \in V} w(x) = M$,
- $p(x, w(x)) = 1$ whenever $x \in I$,
- $q(x, w(x)) = 1$ whenever $x \notin I$?

Motivation for introducing this problem comes from the following observations.

(i) Let (G, w, M) be an instance to the EWIS problem. Let $\{p^x : x \in V\}$ and $\{q^x : x \in V\}$ be defined as in the Step 1 of the above algorithm. Let $p(x, k) := p^x(k)$, $q(x, k) := q^x(k)$ for all $x \in V$ and all k. Then, the solution to EWIS(G, w, M) is YES if and only if the solution to **P1**(G, M, p, q) is YES.

(ii) Let G' be a graph after a certain number of pruning operations in the above algorithm have been performed, with the corresponding functions $\{p^x : x \in V(G')\}$ and $\{q^x : x \in V(G')\}$. Suppose that another pruning step (i.e., Step 3) is applied to G', and let G'' denote the resulting graph, with the corresponding functions $\{p^x_{new} : x \in V(G'')\}$ and $\{q^x_{new} : x \in V(G'')\}$. Let $p(x, k) := p^x(k)$, $q(x, k) := q^x(k)$ for all $x \in V(G')$ and all k, and $p_{new}(x, k) := p^x_{new}(k)$, $q_{new}(x, k) := q^x_{new}(k)$ for all $x \in V(G'')$ and all k. Then, the solution to **P1**(G', M, p, q) is YES if and only if the solution to **P1**$(G'', M, p_{new}, q_{new})$ is YES.

(iii) Letting $p^x := p(x, \cdot)$ and $q^x := q(x, \cdot)$ and running Steps 2-4 of the above algorithm correctly solves the problem **P1**(G, M, p, q).

We leave the routine proofs of these statements to the reader.

The correctness of our algorithm now follows from (i) and (iii). Clearly, the algorithm can be implemented so that it runs in time $O(M^2 n + m)$. □

Let us conclude this subsection by remarking that a slight modification of the algorithm (cf. Cogis and Thierry [6]) would allow us, whenever the solution in YES, to output an independent set that achieves the target weight, as well.

4.6 Circle graphs

Circle graphs are intersection graphs of chords on a circle. In this subsection, we will present a $O(M^2n^2)$ dynamic-programming algorithm for the EWIS problem in circle graphs. Since the class of circle graphs properly contains distance-hereditary graphs, this provides an extension of the result of the previous subsection, at least in term of pseudo-polynomial time solvability of EWIS. Nevertheless, whenever the graph is distance-hereditary, the algorithm from the previous subsection should be applied as it runs in linear time.

Our algorithm for EWIS on circle graphs is based on the dynamic programming solution for the IS problem, developed by Supowit in [23].

Theorem 19. 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 7 in this case.

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

4.7 Partial k-trees

Perhaps the most involved of the dynamic programming algorithms described in this paper is the one for *partial k-trees*, also known as *graphs of treewidth at most k*.

Theorem 20. *For every fixed k,* EWIS *admits an $O(nM^3)$ algorithm for partial k-trees.*

Proof. Let $G = (V, E)$ be a weighted graph and suppose that G is a partial k-tree. First, we construct a special decomposition: let G' be the k-tree containing G as a spanning subgraph, and let $T = (\mathcal{I}, F)$ be a tree-decomposition of G' such that

- all the sets X_i in the decomposition have size $k + 1$,
- if $(i, j) \in F$ then $|X_i \cap X_j| = k$.

It is well-known that such a decomposition can be obtained in linear time from any tree-decomposition of G' of width k.

We further refine this composition by subdividing each edge $[i, j]$ of T, and labeling the new node with the set $X_i \cap X_j$. Now, every edge connects a set of size k with one of its supersets of size $k + 1$.

We root the decomposition tree at an arbitrary node r. The new, rooted decomposition tree has the following properties:

- any node corresponding to a set of size k has exactly one child,
- for a node corresponding to a set X_i of size k, its child corresponds to a superset of X_i of size $k + 1$,
- every child of a node corresponding to a set X_i of size $k+1$ corresponds to a subset of X_i of size k.

For a node i of the decomposition tree, let Y_i denote the set of all vertices of G which appear either in X_i, or in any of the sets corresponding to the descendants of i.

For any node i, any subset Z of X_i, and any integer $p \in \{0, 1, \ldots, M\}$, define the Boolean-valued function

$$ewis(i, Z, w) = \begin{cases} 1, & \text{if there is an independent set } I \text{ in } G[Y_i] \text{ of weight } p \text{ with } I \cap X_i = Z; \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, the answer to the problem EWIS(G, w, M) is YES if and only if there is a subset Z of the set X_r corresponding to the root r such that $ewis(r, Z, M) = 1$.

If X_i is a leaf of the decomposition tree, then it is easy to compute the values $ewis(i, Z, p)$. Indeed, in this case $Y_i = X_i$, so we can set

$$ewis(i, Z, p) = \begin{cases} 1, & \text{if } Z \text{ is an independent set in } G \text{ of weight } p; \\ 0, & \text{otherwise.} \end{cases}$$

For the internal vertices, we consider two cases:

- The size of X_i is k. This implies that i has only one child, say j. The set X_j is a superset of X_i of size $k+1$, so $X_i = X_j \backslash \{v\}$ for some vertex v. Also, $Y_i = Y_j$, since X_i does not add any new vertices. We can compute $ewis(i, Z, p)$ by the following formula

  $$ewis(i, Z, p) = ewis(j, Z, p) \vee ewis(j, Z \cup \{v\}, p)\,,$$

  where $\vee$ denotes the Boolean OR function.

- The size of X_i is $k+1$. Let $\{j_1, \ldots, j_t\}$ be the children of i in the decomposition tree. We would like to compute $ewis(i, Z, p)$, where Z is a subset of X_i, and $w \in \{0, 1, \ldots, M\}$. If Z is not independent, then we set $ewis(i, Z, p)$ to 0.

  From now on, assume that Z is independent. Recall that each of the sets X_{j_s}, for $s \in \{1, \ldots, t\}$, is a subset of X_i.

  Let I be an independent set in $G[Y_i]$ with $I \cap X_i = Z$. Then $I \cap X_{j_s} = Z \cap X_{j_s}$. For $s \in \{1, \ldots, t\}$, let us denote by I_s that part of the set I which belongs to Y_{j_s} but not to Z, i.e., $I_s = I \cap (Y_{j_s} \backslash Z)$. In particular, this implies that $I_s \cap X_i = \emptyset$, and consequently $I_s \cap X_{j_s} = \emptyset$.

  Note that the set I equals to the disjoint union of the set Z and the sets $I_1, \ldots, I_t$. Therefore, if the weight of I is p, then

  $$p = w(I) = w(Z) + \sum_{s=1}^{t} w(I_s)\,.$$

  Thus

  $$\sum_{s=1}^{t} w(I_s) = p - w(Z)$$

  which implies

  $$\sum_{s=1}^{t} w(I \cap Y_{j_s}) = \sum_{s=1}^{t} (w(I_s) + w(Z \cap X_{j_s})) = p - w(Z) + \sum_{s=1}^{t} w(Z \cap X_{j_s})\,.$$

  In particular, $ewis(i, Z, p)$ will take the value 1 if and only if there exist nonnegative integers $p_1, \ldots, p_t$ such that:

  (*i*) $w(Z \cap X_{j_s}) \leq p_s \leq p$ for all $s \in \{1, \ldots, t\}$,

  (*ii*) $\sum_{s=1}^{t} p_s = p - w(Z) + \sum_{s=1}^{t} w(Z \cap X_{j_s})$, and

  (*iii*) $ewis(j_s, Z \cap X_{j_s}, p_s) = 1$ for all $s = 1, \ldots, t$.

  One direction is immediate: if $ewis(i, Z, p)$ takes value 1, then there is an I as above, and we can take $p_s = w(I \cap Y_{j_s})$, for $s \in \{1, \ldots, t\}$.

  On the other hand, the existence of such integers $p_1, \ldots, p_t$ implies that there are sets $I'_1, \ldots, I'_t$ such that, for all $s \in \{1, \ldots, t\}$, I'_s is an independent set in $G[Y_{j_s}]$ of weight p_s with $I'_s \cap X_{j_s} = Z \cap X_{j_s}$.

  We claim that the set $I = \cup_{s=1}^{t} I'_s \cup Z$ is an independent set in $G[Y_i]$ of weight p with $I \cap X_i = Z$. To see this, let us write $I_s = I'_s \cap (Y_{j_s} \backslash Z)$. Then, each I'_s equals to the disjoint union of the sets I_s and $I'_s \cap X_{j_s} = Z \cap X_{j_s}$. Moreover, the set I equals to the disjoint union of the set Z and the sets $I_1, \ldots, I_t$. Therefore,

$$\begin{aligned} p &= \textstyle\sum_{s=1}^t p_s + w(Z) - \sum_{s=1}^t w(Z \cap X_{j_s}) \\ &= w(Z) + \textstyle\sum_{s=1}^t (w(I'_s) - w(Z \cap X_{j_s})) \\ &= w(Z) + \textstyle\sum_{s=1}^t w(I_s) \\ &= w(I)\,. \end{aligned}$$

To see that I intersects X_i in Z, we just need to observe that

$$I \cap X_i = (\cup_{s=1}^t I_s \cup Z) \cap X_i = \cup_{s=1}^t (I_s \cap X_i) \cup (Z \cap X_i) = Z \cap X_i = Z\,,$$

as $I_s \cap X_i = \emptyset$ for all $s \in \{1, \ldots, t\}$.

Finally, we need to show that I is independent. By contradiction, suppose that there are vertices $u, v \in I$ such that $[u, v] \in E(G)$. As Z is independent by assumption, at most one of u and v is contained in Z. We may therefore assume, without loss of generality, that $u \in I_1$. As $I_1 \subseteq I'_1$ and I'_1 is independent, this implies that $v \notin I'_1$.

By tree decomposition properties, there is a set X_{i^*} such that $\{u, v\} \subseteq X_{i^*}$. Also, the set $S_u = \{j : u \in X_j\}$ forms a subtree of our decomposition tree.

Since u is not contained in Z and $I \cap X_i = Z$, the vertex u is not contained in X_i either. However, u is an element of I_1 and is therefore contained in Y_{j_1}.

These observations imply that the set S_u is contained in the subtree rooted at j_1. In particular, the node i^* is a descendant of j_1 in our decomposition tree. Next, it follows from $v \in I \backslash I'_1$ that v is also contained in some X_j such that j is a (not necessarily proper) descendant of i which is *not* contained in the subtree rooted at j_1. As $v \in X_{i^*}$ and as the set $S_v = \{j : v \in X_j\}$ also forms a connected subgraph, we conclude that $v \in X_{j_1}$. However, together with $I \cap X_{j_1} = Z \cap X_{j_1} = I'_1 \cap X_{j_1} \subseteq I'_1$, this leads to a contradicting $v \in I'_1$.

The existence of such p_i's can be determined in time $O(tM^2)$ by dynamic programming. Indeed, finding such w_i's is equivalent to the following restricted version of the GENERALIZED SUBSET SUM problem:

INSTANCE: Nonempty sets of positive integers $A_1, \ldots, A_n$ and a positive integer b.
QUESTION: Are there $a(i) \in A_i$ for all $i \in \{1, \ldots, n\}$ such that $\sum_{i=1}^n a(i) = b$?

Indeed, the correspondence is given by setting

* $n = t$,
* $A_s = \{p_s : w(Z \cap X_{j_s}) \leq p_s \leq p, ewis(j_s, Z \cap X_{j_s}, p_s) = 1\}$ for all $s \in \{1, \ldots, t\}$, and

- $b = p - w(Z) + \sum_{s=1}^{t} w(Z \cap X_{j_s})$.

A slight modification of the proof of Lemma 6 shows that this problem can be solved in time $O(tw^2) = O(tM^2)$.

Summing up over all internal nodes and taking into account that $|E(G)| = O(n)$ (when viewing k as a constant), the total complexity of solving the problem EWIS for partial k-trees is $O(nM^3)$. □

4.8 Graphs of clique-width at most k

The clique-width of a graph G, denoted by $cwd(G)$, 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 $\rho_{i \to j}$).

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.

As shown by Corneil and Rotics in [7], the clique-width of a partial k-tree 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 converse is generally not true, as the complete graphs show. 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.[3] EWIS is no exception.

Theorem 21. *For every fixed k,* EWIS *admits an $O(M^2 l)$ algorithm for graphs of clique-width k, where l is the encoding length (e.g., the number of operations) in a (given) clique-width expression for G.*

[3] 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 [19].

Proof. For every subset of labels $S \subseteq \{1,\ldots,k\}$, let 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 EWIS(G,w,M) by first computing all the values for EWIS(G,w,S,m), for every subset of labels $S \subseteq \{1,\ldots,k\}$, and every $m \in \{0,1,\ldots,M\}$. 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 $S \subseteq \{1,\ldots,k\}$, and for all $m \in \{0,1,\ldots,M\}$, let

$$\mathrm{EWIS}(G,w,S,m) = \begin{cases} \text{YES}, & \text{if } S = \{label(v)\} \text{ and } m \in \{0, w(v)\};\\ \text{YES}, & \text{if } S = \emptyset \text{ and } m = 0;\\ \text{NO}, & \text{otherwise.}\end{cases}$$

If $G = G_1 \oplus G_2$ then let for all $S \subseteq \{1,\ldots,k\}$, and for all $m \in \{0,1,\ldots,M\}$:

$$\mathrm{EWIS}(G,w,S,m) = \begin{cases} \text{YES}, & \text{if } \mathrm{EWIS}(G_1,w,S,m) = \text{YES};\\ \text{YES}, & \text{if } \mathrm{EWIS}(G_2,w,S,m) = \text{YES};\\ \text{YES}, & \text{if there is an } m' \in \{0,1,\ldots,m\} \text{ such that}\\ & \quad \mathrm{EWIS}(G_1,w,S,m') = \mathrm{EWIS}(G_2,w,S,m-m') = \text{YES};\\ \text{NO}, & \text{otherwise.}\end{cases}$$

This can be computed in time $O(M^2)$, similarly as in Corollary 7.
If $G = \eta_{i,j}(G_1)$ then let for all $S \subseteq \{1,\ldots,k\}$, and for all $m \in \{0,1,\ldots,M\}$:

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

If $G = \rho_{i\to j}(G_1)$ then let for all $S \subseteq \{1,\ldots,k\}$, and for all $m \in \{0,1,\ldots,M\}$:

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

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

$$\mathrm{EWIS}(G,w,M) = \begin{cases} \text{YES}, & \text{if } \exists\, S \subseteq \{1,\ldots,k\} \text{ such that } \mathrm{EWIS}(G,w,S,M) = \text{YES};\\ \text{NO}, & \text{otherwise.}\end{cases}$$

□

Let us close this section by remarking that all of the above algorithms can be easily modified so that an independent set achieving the target weight is also produced (whenever there is one).

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. As the exact matching problem for bipartite graphs is the same as the exact weighted independent set problem for line graphs of bipartite graphs, 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 $L(B)$ of line graphs of bipartite graphs is sandwiched between two graph classes for which the complexity of EWIS is known, and whose sets of forbidden induced subgraphs differ only in two graphs.

Line graphs of bipartite graphs form a hereditary class of graphs. Their characterization in terms of forbidden induced subgraphs has been obtained in [22], as follows. A graph G is the line graph of a bipartite graph if and only G is F-free, where $F = \{K_{1,3}, \mathit{diamond}, C_5, C_7, \ldots\}$. A *diamond* is the graph obtained by deleting a single edge from a complete graph on 4 vertices.

Replacing the diamond in the above characterization by its subgraph C_3 results in a smaller class of $(K_{1,3}, C_3, C_5, C_7, \ldots)$-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 7 and Theorem 20, the problem is solvable in pseudo-polynomial time in this class.

On the other hand, if we replace $K_{1,3}$ with $K_{1,4}$ in the above characterization of $L(B)$, we obtain a class of graphs that strictly contains line graphs of bipartite graphs. This class of $(K_{1,4}, \mathit{diamond}, C_5, C_7, \ldots)$-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 Sections 1 and 3.1 imply that the problem is strongly **NP**-complete for this class, and hence also for the larger class of $(K_{1,4}, \mathit{diamond}, C_5, C_7, \ldots)$-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.

$$\begin{array}{ccccc} \mathit{Free}(\{K_{1,3}, C_3, C_5, C_7, \ldots\}) & \subset & L(Bip) & \subset & \mathit{Free}(\{K_{1,4}, \mathit{diamond}, C_5, C_7, \ldots\}) \\ \text{pseudo-polynomial} & & ??? & & \text{strongly NP-complete} \end{array}$$

References

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

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

[3] C. Berge (1973). Graphs and hypergraphs. *North Holland, Amsterdam.*

[4] J. Błażewicz, P. Formanowicz, M. Kasprzak, P. Schuurman and G. Woeginger, "DNA Sequencing, Eulerian Graphs, and the Exact Perfect Matching Problem," Graph-theoretic concepts in computer science, 13–24, *Lecture Notes in Comput. Sci.* **2573**, Springer, Berlin, 2002.

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

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

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

[8] 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.

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

[10] V.G. Deĭneko and G.J. Woeginger, "On the robust assignment problem under a fixed number of cost scenarios," *Operations Research Letters* **34** (2006) 175–179.

[11] R. Diestel, Graph theory. Third edition. *Graduate Texts in Mathematics*, 173. Springer-Verlag, Berlin, 2005.

[12] M. Farber, M. Hujter and Zs. Tuza, "An upper bound on the number of cliques in a graph," *Networks* **23** (1993) 207-210.

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

[14] 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.

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

[16] V. Lozin and M. Milanič, "Maximum Independent Sets in Graphs of Low Degree," to appear in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms SODA'07.*

[17] C. Mannino, G. Oriolo, F. Ricci and S. Chandran. "The stable set problem and the thinness of a graph," *Operations Research Letters*, In Press.

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

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

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

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

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

[23] K.J. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **CAD-6** (1987) 93–94.

[24] R. Thomas, "A survey of Pfaffian orientations of graphs," to appear in *Proceedings of the International Congress of Mathematicians, Madrid, Spain, 2006.*

[25] 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.