

Chapitre 1

On the complexity of the exact weighted independent set problem

1.1. 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 [PAPADIMITRIOU 82] 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 [BARAHONA 87].

– The *exact cut* problem is solvable in pseudo-polynomial time for planar and toroidal graphs [BARAHONA 87].

– The *exact perfect matching* problem is solvable in pseudo-polynomial time for planar graphs [BARAHONA 87], and more generally, for graphs that have a Pfaffian orientation¹ (provided one is given). We recall that a *matching* of a graph $G = (V, E)$ is a set E' of pairwise non-adjacent edges of G . If $|E'| = |V|/2$, then E' is said to be a *perfect matching* of G . Karzanov [KARZANOV 87] gives a polynomial-time

Chapitre rédigé par Martin MILANIČ et Jérôme MONNOT.

1. Bipartite graphs with a Pfaffian orientation have been characterized in [THOMAS 06], where a polynomial-time recognition algorithm is also presented.

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 [PAPADIMITRIOU 82] 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 [MULMULEY 87] show that the exact perfect matching problem has a randomized pseudo-polynomial-time algorithm.

The exact perfect matching problem is of great practical importance. It has applications in such diverse areas as bus-driver scheduling, statistical mechanics (see [LECLERC 86]), DNA sequencing [BŁAŻEWICZ 06], and robust assignment problems [DEJNEKO 06]. The problem consists in determining whether a given edge-weighted graph contains a perfect matching of a given weight. Despite polynomial results for special cases, the deterministic complexity of the exact perfect matching problem remains unsettled for general graphs, and even for bipartite graphs. Papadimitriou and Yannakakis conjectured that the problem is **NP**-complete [PAPADIMITRIOU 82].

This open problem 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). 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 $w : V \rightarrow \mathbb{Z}$ contains an independent set whose total weight (i.e., the sum of the weights of its members) equals a given integer b . Formally, the solution to the EWIS problem, given (G, w, b) , is *yes* if and only if there is an independent set I of G with $w(I) = b$. The restriction where we require the independent set to be of a maximum independent set of the graph will be called the *exact weighted maximum independent set problem* and denoted by EWIS_α . Thus, $\text{EWIS}_\alpha(G, w, b)$ asks about the existence of an independent set I of G with $|I| = \alpha(G)$ and $w(I) = b$.

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. The *exact matching* problem, that is, the problem of determining whether a given edge-weighted graph contains a matching of a given

weight, is then 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.

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

More specifically, we can summarize the main results of this chapter 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* 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 in each of the following graph classes :

- mK_2 -free graphs,
- interval graphs, and their generalizations k -thin graphs,
- circle graphs,
- chordal graphs,
- AT-free graphs,
- (claw, net)-free graphs,
- distance-hereditary graphs,
- graphs of bounded treewidth,
- graphs of bounded clique-width,
- certain subclasses of P_5 -free and fork-free graphs.

The results for subclasses of P_5 -free and fork-free graphs are derived by means of modular decomposition. The application of modular decomposition to the EWIS problem is described in Section 1.4.2 and may be of independent interest.

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 the EWIS_α problem is (pseudo-) polynomially solvable for a class of graphs \mathcal{G} , the exact

2. We also strengthen this result considerably, however we postpone the detailed formulation until Section 1.3.2.

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.

We say that a graph H is an *induced subgraph* of G if H can be obtained from G by deletion of some (possibly none) vertices ; the subgraph of G *induced by* $U \subseteq V(G)$ is the graph obtained from G by deleting the vertices from $V(G) \setminus U$ and it will be denoted by $G[U]$. For a graph G , we denote by $\text{co-}G$ (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 [BERGE 73]. 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 $\{1, \dots, k\}$.

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

The chapter is organized as follows. In Section 1.2, 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_α . Section 1.3 is devoted to the strong **NP**-completeness results. In Section 1.4, pseudo-polynomial time solutions to the exact weighted independent set problem are presented. We conclude the chapter with a short discussion in Section 1.5 that places the class of line graphs of bipartite graphs between two graph classes with known complexities of the EWIS problem.

1.2. Preliminary observations

The exact weighted independent set problem is (weakly) **NP**-complete for any class of graphs containing the edgeless graphs $\{\overline{K}_n : n \geq 0\}$. 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 [GAREY 79]). The *subset sum* problem is the following : given n integers a_1, \dots, a_n and a bound b , determine whether there is a subset $J \subseteq [n]$ such that $\sum_{j \in J} a_j = b$.

Therefore, for a given class of graphs \mathcal{G} , the question of interest is whether the EWIS problem is strongly **NP**-complete for graphs in \mathcal{G} , or is it solvable in pseudo-polynomial time.

First, let us observe that in any class of graphs \mathcal{G} , we may restrict our attention to instances with positive weights only.

REMARQUE 1.– The EWIS problem with arbitrary integer weights is polynomially equivalent to the EWIS problem, restricted to instances (G, w, b) such that $b \leq w(V(G))$ and $1 \leq w(v) \leq b$, for all $v \in V$. The same equivalence holds true for the EWIS_α problem.

PREUVE.– 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 $b = 0$, in which case the solution is trivial, as the empty set is an independent set of weight 0). 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 b by $b + kN$ 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 b , as they will never appear in a solution. Furthermore, the solution is clearly *no* if $b > w(V)$.

The same assumption on vertex weights as for the EWIS problem can also be made for the instances (G, w, b) of its restricted counterpart EWIS_α . 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 value of $\text{EWIS}_\alpha(G, \mathbf{1}, p)$ 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_α , when restricted to particular graph classes.

LEMME 1.1.– *Let \mathcal{G} be a class of graphs. The following statements are true.*

- (i) *If the EWIS_α problem is solvable in pseudo-polynomial time for graphs in \mathcal{G} , then the WIS problem is solvable in pseudo-polynomial time for graphs in \mathcal{G} .*
- (ii) *If the EWIS problem is solvable in pseudo-polynomial time for graphs in \mathcal{G} , then the EWIS_α problem 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 the EWIS_α problem is solvable in pseudo-polynomial time for graphs in \mathcal{G}' , then the EWIS problem is solvable in pseudo-polynomial time for graphs in \mathcal{G} .*

PREUVE.– (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 b from $w(V)$ down to k and using an algorithm for the EWIS_α problem on the instance (G, w, b) , we can decide whether G contains a maximum independent set of total weight at least k .

(ii) Let (G, w, b) 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 $k \in [n]$ such that the value of $\text{EWIS}(G, \mathbf{1}, k)$ 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 $b' = b + \alpha(G)N$. Then it is easy to verify that $\text{EWIS}_\alpha(G, w, b) = \text{EWIS}(G, w', b')$.

(iii) Let (G, w, b) 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 value of $\text{EWIS}(G, w, b)$ is *yes* if and only if the value of $\text{EWIS}_\alpha(G', w', b')$ is *yes* for some $b' \in \{(n + 1)b, \dots, (n + 1)b + n - 1\}$.

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, Lemme 1.1 implies the following result.

COROLLAIRE 1.1.– *Let \mathcal{G} be a class of graphs. The following statements are true.*

- (i) *If the WIS problem is strongly **NP**-complete for graphs in \mathcal{G} , then the EWIS_α problem is strongly **NP**-hard for graphs in \mathcal{G} . If, in addition, the IS problem is polynomial for graphs in \mathcal{G} , then the EWIS_α problem is strongly **NP**-complete for graphs in \mathcal{G} .*
- (ii) *If the EWIS_α problem is strongly **NP**-hard for graphs in \mathcal{G} , then the EWIS problem is strongly **NP**-complete for graphs in \mathcal{G} .*
- (iii) *Let \mathcal{G}' be as in Lemme 1.1. If the EWIS problem is strongly **NP**-complete for*

graphs in \mathcal{G} , then the EWIS_α problem is strongly **NP**-hard for graphs in \mathcal{G}' . If, in addition, the IS problem is polynomial for graphs in \mathcal{G}' , then the EWIS_α problem 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 $\mathcal{G} \in \{\text{forests, bipartite graphs, chordal graphs}\}$, 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). 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 a similar equivalence remains valid for the class of line graphs. More precisely, if L , $L(\text{Bip})$, $L(K_{2n})$ 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.

LEMME 1.2.– *The EWIS problem is strongly **NP**-complete for graphs in L (resp., $L(\text{Bip})$) if and only if the EWIS_α problem is strongly **NP**-complete for graphs in $L(K_{2n})$ (resp., $L(K_{n,n})$).*

PREUVE.– The backward implication is given by part (ii) of Lemme 1.1. 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 b for the exact matching problem, construct an instance $(K_{n'}, w', b')$ 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 b if and only if $K_{n'}$ has a perfect matching of weight $Nb + k$ 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.

1.3. 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.

1.3.1. 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 connects 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 the EWIS problem in bipartite graphs follows from a straightforward reduction from the balanced biclique problem which is known to be **NP**-complete [GAREY 79]. The *balanced biclique* problem consists in determining whether, given a bipartite graph $G = (L, R; E)$ and an integer k , there exist subsets $L' \subseteq L$ 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 [DAWANDE 01], 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 $b = k + Bk$. It is clear that there exist an independent set in $(L, R; (L \times R) \setminus E)$ with weight b if and only if there exists a balanced biclique in $(L, R; E)$ of size k .

We now strengthen this result by proving that the EWIS_α problem is strongly **NP**-complete for cubic bipartite graphs. By contrast, for graphs of maximum degree 2, EWIS and EWIS_α 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 2.³ By Corollaire 1.3 and Théorème 1.14 from Section 1.4, the problem is solvable in pseudo-polynomial time in this class.

THÉORÈME 1.1.– *The EWIS_α problem is strongly **NP**-complete in cubic bipartite graphs.*

PREUVE.– 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 [GAREY 79]. A *clique* V^* is a subset of vertices of G such that the subgraph induced by V^* is complete. Let (G, k) be an input to the clique problem, where $G = (V, E)$ is a Δ -regular graph on 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 the EWIS_α problem where $G' = (L, R; E')$ is a bipartite graph as follows :

3. For the definition of treewidth, see Section 1.4.1.8.

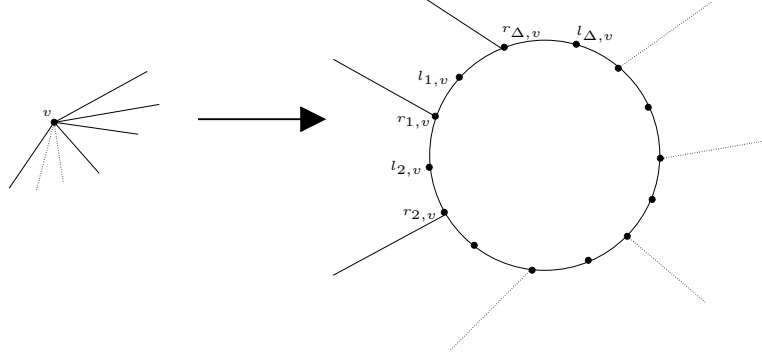


Figure 1.1. The gadget $H(v)$.

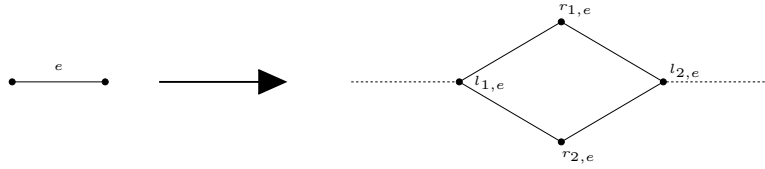


Figure 1.2. The gadget $H(e)$.

- 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 $L_v = \{l_{1,v}, \dots, l_{\Delta,v}\}$ and the right set is $R_v = \{r_{1,v}, \dots, r_{\Delta,v}\}$. The weights are $w(l_{i,v}) = 1$ and $w(r_{i,v}) = n\Delta \binom{2+n\Delta}{2}$ for $i \in [\Delta]$. The gadget $H(v)$ is illustrated in Figure 1.1.

- For each edge $e \in E$, we construct a gadget $H(e)$ which is 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 $w(r_{i,e}) = \frac{(n\Delta)^2}{2} \binom{2+n\Delta}{2}$ for $i = 1, 2$. The gadget $H(e)$ is illustrated in Figure 1.2.

- 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 the value of $\text{EWIS}_\alpha(G', w, b)$ is *yes*, where

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

Let I be a maximum independent set of G' with $w(I) = b$. 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). By construction of the weights, the quantity $k\Delta$ of b 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 :

$$|I \cap L_V| = k\Delta, \quad |I \cap R_V| = (n-k)\Delta. \quad [1.1]$$

where $L_V = \cup_{v \in V} L_v$ and $R_V = \cup_{v \in V} R_v$. Thus, using (1.1) we must obtain :

$$w(I \cap (L_E \cup R_E)) = n\Delta \frac{k(k-1)}{2} + n\Delta \left(\frac{2+n\Delta}{2} \right) \left(\frac{n\Delta}{2} - \frac{k(k-1)}{2} \right) n\Delta. [1.2]$$

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 $k \binom{k-1}{2}$ gadgets $H(e)$ with $L_e \subseteq I$. Assume the converse ; 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 (1.2), we deduce that $p = 0$, contradiction.

Thus, if we set $V^* = \{v \in V : L_v \subseteq I\}$, we deduce from above that $|V^*| = k$ 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 $w(S) = b$ and that S 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 S , we deduce that $e = [u, v] \in E(V^*)$ and then $L_v \subseteq I$, contradiction. The proof is complete.

As corollary of Théorème 1.1, 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 Théorème 1.1 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 Théorème 1.1 implies the strong **NP**-completeness of the EWIS_α problem for perfect graphs, a well-known class where the weighted independent set problem is solvable in polynomial time [GRÖTSCHEL 84].

1.3.2. A more general hardness result

Let us now strengthen the main result of the previous subsection. To this end, we first introduce some notations. Let \mathcal{F} be a set of graphs. We denote the class of graphs containing no induced subgraphs from the set \mathcal{F} by $\text{Free}(\mathcal{F})$. Any graph in $\text{Free}(\mathcal{F})$ 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} .

Let C_i and H_i denote the cycle of length i and the graph in Figure 1.3, respectively.

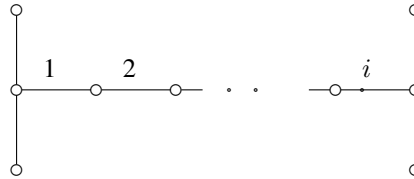


Figure 1.3. Graph H_i

We associate to every graph G a parameter $\kappa(G)$, which is the minimum value of $i \geq 1$ such that G contains an induced copy of either C_i or H_i . If G is an acyclic graph with no induced graphs of the form H_i , we let $\kappa(G) = \infty$. For a (possibly infinite) nonempty set of graphs \mathcal{F} , we define

$$\kappa(\mathcal{F}) = \sup \{ \kappa(G) : G \in \mathcal{F} \}.$$

Finally, for a set of graphs X , let X_3 denote the set of graphs of degree at most 3 in X .

With these definitions in mind, we can use the strong **NP**-completeness of the EWIS_α problem for bipartite graphs of degree at most 3 (which is an immediate corollary of Théorème 1.1), and the reduction typically used for the IS problem (see e.g. [MURPHY 92, POLJAK 74]), to derive the following hardness result.

THÉORÈME 1.2.– *Let \mathcal{G} be the class of \mathcal{F} -free bipartite graphs of maximum degree at most 3. If $\kappa(\mathcal{F}_3) < \infty$, then the EWIS_α problem is strongly **NP**-complete in the class \mathcal{G}_3 .*

PREUVE.– 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 $(C_3, \dots, C_k, H_1, \dots, H_k)$ -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, b) be an instance of the EWIS_α problem 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 value of $\text{EWIS}_\alpha(G, w, b)$ is *yes* if and only if the value of $\text{EWIS}_\alpha(G', w', b + mk'N)$ 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 value of $\text{EWIS}_\alpha(G', w', b + mk'N)$ is *yes*. Let I' be a maximum independent set of G' of weight $b + mk'N$. 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 $w(V) + (mk' - 1)N$, 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' would 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 b , we conclude that the value of $\text{EWIS}_\alpha(G, w, b)$ is *yes*.

This shows that the EWIS_α problem is strongly **NP**-complete in the class \mathcal{S}_k . To prove strong **NP**-completeness of the problem in the class \mathcal{G}_3 , we now show that the class \mathcal{G}_3 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}_3 . Then G contains a graph $A \in \mathcal{F}_3$ as an induced subgraph. From the choice of G we know that A belongs to \mathcal{S}_k , but then $k < \kappa(A) \leq \kappa(\mathcal{F}_3) \leq k$, a contradiction. Therefore, $G \in \mathcal{G}_3$ and the theorem is proved.

1.4. 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.⁴ The algorithms resemble those for the WIS problem in respective graph classes, and are based either on a dynamic programming approach (Section 1.4.1), or on the modular decomposition (Section 1.4.2).

First, we observe that when developing polynomial-time solutions to the EWIS problem, we may restrict our attention to connected graphs.

LEMME 1.3.— *Let (G, w, b) be an instance of the EWIS problem, and let C_1, \dots, C_r be the connected components of G . Suppose that for each $i \in [r]$, the set of solutions $(\text{EWIS}(C_i, w, k) : k \in [b])$ for C_i is given. Then, we can compute the set of solutions $(\text{EWIS}(G, w, k) : k \in [b])$ for G in time $O(rb^2)$.*

In order to show Lemme 1.3, we consider the following generalization of the subset sum problem.

GENERALIZED SUBSET SUM (GSS)

Instance : Nonempty sets of positive integers A_1, \dots, A_n and a positive integer b .

Question : Is there a nonempty subset J of $[n]$ and a mapping $a : J \rightarrow \cup_{j \in J} A_j$ such that $a(j) \in A_j$ for all $j \in J$, and $\sum_{j \in J} a(j) = b$?

By generalizing the dynamic programming solution to the subset sum problem, it is easy to show the following.

LEMME 1.4.— *Generalized subset sum can be solved in time $O(nb^2)$ by dynamic programming.*

In fact, in the stated time, not only we can verify if there is a $J \subseteq [n]$ and a mapping a as above such that $\sum_{j \in J} a(j) = b$ for the given b , but we can answer this question for *all* values $b' \in [b]$.

PREUVE.— Let B denote the set of *all* values $b' \in [b]$ such that there a nonempty subset S of $[n]$ and a mapping $a : S \rightarrow \cup_{i \in S} A_i$ such that $a(i) \in A_i$ for all $i \in S$, and $\sum_{i \in S} a(i) = b'$.

Let us show by induction on n that we can generate the set B in time $O(nb^2)$. The statement is trivial for $n = 1$.

Suppose now that $n > 1$. Let $I = (A_1, \dots, A_n; b)$ be an instance of the GSS problem. Let B' be the inductively constructed set of all possible values of $b' \in [b]$

4. For a comprehensive survey on graph classes, see [BRANDSTÄDT 99].

such that the solution to the GSS problem on the instance $(A_1, \dots, A_{n-1}; b')$ 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 problem, given $(A_1, \dots, A_n; \beta)$, 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 above statement and hence the lemma.

Lemme 1.3 now follows immediately.

PREUVE.– [Lemme 1.3] It is enough to observe that for every $k \in [b]$, the value of $\text{EWIS}(G, w, k)$ is *yes* if and only if the solution to the GSS problem on the instance $(A_1, \dots, A_r; k)$ is *yes*, where A_i denotes the set of all values $k' \in [b]$ such that $\text{EWIS}(C_i, w, k')$ is *yes*.

1.4.1. Dynamic programming solutions

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

THÉORÈME 1.3.– *The exact weighted independent set and the exact weighted maximum independent set problems admit pseudo-polynomial-time solutions in each of the following graph classes : mK_2 -free graphs, interval graphs and their generalizations k -thin graphs, circle graphs, chordal graphs, AT-free graphs, (claw, net)-free graphs, distance-hereditary 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 Lemme 1.1, it suffices to develop pseudo-polynomial solutions for the EWIS problem. Most of the algorithms resemble those for the WIS problem and exploit the special structure of graphs in the classes.

1.4.1.1. 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.

THÉORÈME 1.4.– *For every positive integer m , the EWIS problem admits a pseudo-polynomial algorithm for mK_2 -free graphs.*

PREUVE.– An mK_2 -free graph contains only polynomially many maximal independent sets (see [ALEKSEEV 91, BALAS 89, PRISNER 95]). Tsukiyama *et al.* describe in [TSUKIYAMA 77] 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, \dots, I_N of a given mK_2 -free graph G in polynomial time.

For each vertex x of G and for each $k \in [b]$, the value of $\text{EWIS}(G[\{x\}], w, k)$ is *yes* if and only if $k = w(x)$. Therefore, we can apply Corollaire 1.3 to each maximal independent set I of G in order to compute the set of solutions to $\text{EWIS}(G[I], w, k)$ for all $k \in [b]$ in pseudo-polynomial time.

As each independent set of G is contained in some maximal independent set I_i of G , the value of $\text{EWIS}(G, w, k)$ is *yes* if and only if there is an $i \in [N]$ such that the value of $\text{EWIS}(I_i, w, k)$ is *yes*. This shows that the EWIS problem can be solved in pseudo-polynomial time for mK_2 -free graphs.

1.4.1.2. 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} 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 two intervals if and only if their intersection is nonempty. The collection \mathcal{I} is said to be an *interval model* of $G(\mathcal{I})$. Finally, 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 [RAMALINGAM 88] that a graph $G = (V, E)$ is interval if and only if it admits a vertex ordering (v_1, \dots, v_n) such that for all triples (r, s, t) with $1 \leq r < s < t \leq n$, the following implication is true :

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

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.

THÉORÈME 1.5.– *The EWIS problem admits an $O(bn + m)$ algorithm for interval graphs.*

PREUVE.– Let (v_1, \dots, 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 $\{v_1, \dots, v_i\}$ (also, let G_0 be the empty graph). Then, for every $i \in [n]$, either there is a $j = j(i)$ such that $N_{G_i}(v_i) = \{j, j+1, \dots, i-1\}$, 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 $I \setminus \{v_i\}$ 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 $O(bn + m)$ dynamic programming solution to the EWIS problem on interval graphs.

Step 1. Find a vertex ordering (v_1, \dots, v_n) as above.

Step 2. Set $\text{EWIS}(G_0, w, k)$ to *no* for all $k \in [b]$.

Step 3. For $i = 1, \dots, n$, do the following :

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

3.2. For $k \in [b]$, do the following :

If $k = w(v_i)$, set $\text{EWIS}(G_i, w, k)$ to *yes*.

If $k < w(v_i)$, set $\text{EWIS}(G_i, w, k)$ to $\text{EWIS}(G_{i-1}, w, k)$.

If $k > w(v_i)$, set $\text{EWIS}(G_i, w, k)$ to *yes* if at least one of the solutions to $\text{EWIS}(G_{j(i)-1}, w, k - w(v_i))$ and $\text{EWIS}(G_{i-1}, w, k)$ is *yes*, and to *no* otherwise.

Step 4. Output the value of $\text{EWIS}(G_n, w, b)$.

1.4.1.3. k -thin graphs

The property used in the above characterization of interval graphs has been generalized by Mannino *et al.* in [MANNINO 07], 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 $1 \leq r < s < t \leq n$, 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 [MANNINO 07].

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.

THÉORÈME 1.6.— *Suppose that for a k -thin graph $G = (V, E)$, $k \geq 2$, 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 $1 \leq r < s < t \leq n$, 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, the EWIS problem admits an $O(bn^k)$ algorithm for G .*

PREUVE.— The proof is a straightforward extension of the proof of Théorème 1.5. 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 $(i_1, \dots, i_k) \in [|V_1|] \times \dots \times [|V_k|]$.

1.4.1.4. 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 will present a $O(b^2 n^2)$ dynamic-programming algorithm for the EWIS problem in circle graphs.

Our algorithm for the EWIS problem on circle graphs is based on the dynamic programming solution for the IS problem, developed by Supowit in [SUPOWIT 87].

THÉORÈME 1.7.— *The EWIS problem admits an $O(b^2 n^2)$ algorithm for circle graphs.*

PREUVE.— 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 $\{i, i+1, \dots, j\}$. Obviously $G = G(1, 2N)$.

Let $1 \leq i < j \leq 2N$. If $j = i + 1$ then the value of $\text{EWIS}(G(i, j), w, k)$ 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 value of $\text{EWIS}(G(i, j), w, k)$ equals to the value of $\text{EWIS}(G(i, j - 1), w, k)$. 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 $G(i, j - 1)$ 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 value of $\text{EWIS}(G(i, j), w, k)$ is *yes* if and only if either $\text{EWIS}(G(i, j - 1), w, k)$ is *yes*, or $\text{EWIS}(G', w, k)$ 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 Corollaire 1.3 in this case.

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

1.4.1.5. 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 ([FRANK 76, TARJAN 85]) 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 characterizations of chordal graphs : their *clique-tree representation*.

THÉORÈME 1.8.– *The EWIS problem admits an $O(b^2n(n+m))$ algorithm for chordal graphs.*

PREUVE.– Given a chordal graph G , we first compute a *clique tree* of G . This can be done in time $O(n+m)$ [HSU 99]. 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 [OKAMOTO 05], 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, \dots, 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 [OKAMOTO 05], for every distinct $i, j \in [l]$, the sets $V(G(K_i)) \setminus K$ and $V(G(K_j)) \setminus K$ are disjoint. Moreover, if \sqcup denotes the disjoint union, the following relations hold :

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

We extend our usual Boolean predicate $\text{EWIS}(H, w, k)$ to the following two : for a vertex v of a weighted graph (H, w) and an integer k , let $\text{EWIS}(H, w, k, v)$ denote the Boolean predicate whose value 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 $\overline{\text{EWIS}}(H, w, k, U)$ 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 :

$$\begin{aligned}
 \text{EWIS}(G(K), w, k) &= \overline{\text{EWIS}}(G(K), w, k, K) \\
 &\quad \vee \bigvee_{v \in K : w(v) \leq k} \text{EWIS}(G(K), w, k, v). \quad [1.3]
 \end{aligned}$$

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.

$$\text{EWIS}(G(K), w, k, v) = \mathbf{GSS}(A_1, \dots, A_l, k - w(v)) \quad [1.4]$$

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

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

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 $I_i \setminus \{v\}$ (over all $i \in [l]$) equals to the weight of $\left(\bigcup_{i=1}^l I_i\right) \setminus \{v\}$, thus justifying Equation [1.4].

Similarly, we have

$$\overline{\text{EWIS}}(G(K), w, k, K) = \mathbf{GSS}(A_1, \dots, A_l, k) \quad [1.5]$$

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

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

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

$$\overline{\text{EWIS}}(G(K_i), w, k, K \cap K_i) = \overline{\text{EWIS}}(G(K_i), w, k, K_i) \vee \bigvee_{u \in K_i \setminus K} \text{EWIS}(G(K_i), w, k, u). \quad [1.6]$$

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

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

and

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

For every other node K , we compute the values of $\overline{\text{EWIS}}(G(K), w, k, K)$ and $\text{EWIS}(G(K), w, k, v)$ by referring to the recursive relations [1.6], [1.5] and [1.4] in this order. Finally, the value of $\text{EWIS}(G, w, k)$ equals to the value of $\text{EWIS}(G(K_r), w, k)$, which can be computed using Equation [1.3].

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, \dots, K_l , the number of operations performed is $O(\sum_{i=1}^l |K_i| + lb^2 + |K|lb^2)$. 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 $\sum_{K \in \mathcal{V}(T)} |K| = O(n + m)$ [OKAMOTO 05], the claimed complexity bound follows.

1.4.1.6. *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 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 $G - N[x]$.⁵ A graph is called *AT-free* if it has no asteroidal triples.

Our dynamic programming algorithm that solves the EWIS problem 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 [BROERSMA 99].

Let us start with a definition.

DÉFINITION 1.1.— *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 $V(G) \setminus \{x, y\}$ 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 [BROERSMA 99].

THÉORÈME 1.9.— [BROERSMA 99] *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, \dots, C_t^s of $G - N[s]$ such that the components of $I \setminus N[s]$ are precisely $I(x, s)$, $I(s, y)$, and C_1^s, \dots, C_t^s .*

THÉORÈME 1.10.— [BROERSMA 99] *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 $N[D] \cap (N[x] \setminus N[y]) = \emptyset$ if and only if D is a component of $G - N[y]$.*

THÉORÈME 1.11.— [BROERSMA 99] *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, \dots, 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.

LEMME 1.5.— *Let (G, w) be a weighted graph. Then, the value of $\text{EWIS}(G, w, k)$ is yes if and only if there is a vertex $x \in V(G)$ such that the value of $\text{EWIS}(G - N(x), w, k)$ is yes.*

Combining Lemme 1.5 with Théorèmes 1.10 and 1.11, we obtain the following result.

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

LEMME 1.6.— *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 value of $\text{EWIS}(C, w, k)$ is yes if and only if there is a vertex $y \in C$ such that the value of $\text{EWIS}(C_y, w, k)$ 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 Lemme 1.5 with Théorème 1.9, we obtain the following result.

LEMME 1.7.— *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 value of $\text{EWIS}(G[I], w, k)$ 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 value of $\text{EWIS}(I, w, k)$ is yes if and only if there is a vertex $s \in I$ such that the value of $\text{EWIS}(I_s, w, k)$ 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 .*

THÉORÈME 1.12.— *The EWIS problem admits a pseudo-polynomial algorithm for AT-free graphs.*

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

Step 1. For every $x \in V$, compute the 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 $\text{EWIS}(C, w, k)$, for each component C (for all $k \in [w(C)]$) and the solutions to $\text{EWIS}(I, w, k)$ for each interval I (for all $k \in [w(I)]$). To compute the solutions to $\text{EWIS}(C, w, k)$ for a component C , first compute the solutions to $\text{EWIS}(C - N(y), w, k)$, for all $y \in C$, by applying Lemme 1.6 and Corollaire 1.3. Similarly, to compute the solutions to $\text{EWIS}(I, w, k)$ for an interval I , first compute the solutions to $\text{EWIS}(I - N(s), w, k)$, for all $s \in I$, by applying Lemme 1.7 and Corollaire 1.3.

Step 5. Compute $\text{EWIS}(G, w, b)$ using Lemme 1.5 and Corollaire 1.3.

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. The following result is an immediate consequence of Théorème 1.12.

COROLLAIRE 1.2.— *The EWIS problem admits a pseudo-polynomial algorithm for (claw, net)-free graphs.*

PREUVE.— In [BRANDSTÄDT 03], 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, the problem reduces to solving $O(nb)$ subproblems in *AT*-free graphs, which can be done in pseudo-polynomial time by Théorème 1.12.

1.4.1.7. 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.⁶ Bandelt and Mulder provided in [BANDELDT 86] 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 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 $\sigma = (x_1 R_1 y_1, x_2 R_2 y_2, \dots, x_{n-1} R_{n-1} y_{n-1})$ where (x_1, \dots, x_n) is a total ordering of $V(G)$ such that for all $i \in [n - 1]$, the following holds :

- $R_i \in \{P, T, F\}$.
- If we denote by G_i the subgraph of G induced by $\{x_i, \dots, x_n\}$, 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 [DAMIAND 01]. Our algorithm for the EWIS problem on distance-hereditary graphs is based on the dynamic programming solution for the WIS problem, developed by Cogis and Thierry in [COGIS 05].

We remark that that every distance-hereditary graph is a circle graph. However, the algorithm developed here for distance-hereditary graphs is faster than the one for general circle graphs given by Théorème 1.7.

THÉORÈME 1.13.— *The EWIS problem admits an $O(b^2n+m)$ algorithm for distance-hereditary graphs.*

6. 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.

PREUVE.— We first define an auxiliary problem :

$P1(G, b, p, q)$

Instance : A graph G , a positive integer b , and two functions

$$p, q : V \times \{0, 1, \dots, b\} \rightarrow \{0, 1\}.$$

Question : Is there an independent set I of G , and a mapping $w : V \rightarrow \{0, 1, \dots, b\}$ such that the following holds :

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

Let us show that the EWIS problem is reducible to the P1 problem in $O(nb)$ time. Let (G, w, b) be an instance to the EWIS problem. Define $p, q : V(G) \times \{0, 1, \dots, b\} \rightarrow \{0, 1\}$ as follows. For each $x \in V(G)$ and each $k \in \{0, 1, \dots, b\}$, let

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

and

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

Then, it is easy to see that the value of $EWIS(G, w, b)$ is *yes* if and only if $P1(G, b, p, q)$ is *yes*.

In what follows, we present an $O(b^2n + m)$ to solve the problem P1 on an instance (G, b, p, q) , if G is a distance-hereditary graph. This will in turn imply the statement of the theorem. For two functions $f, g : \{0, 1, \dots, N\} \rightarrow \{0, 1\}$, we denote their *convolution* $f * g$ as the function $f * g : \{0, 1, \dots, N\} \rightarrow \{0, 1\}$, given by the following rule : for every $k \in \{0, 1, \dots, N\}$, we have

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

Procedure P1-DH

Input : A distance-hereditary graph G , a positive integer b , and two functions

$$p, q : V \times \{0, 1, \dots, b\} \rightarrow \{0, 1\}.$$

Output : The answer to the question in $P1(G, b, p, q)$.

Step 1. Compute the pruning sequence $\sigma = (x_1 R_1 y_1, x_2 R_2 y_2, \dots, x_{n-1} R_{n-1} y_{n-1})$ for G . To each vertex $x \in V(G)$, associate a pair of functions $p^x, q^x : \{0, 1, \dots, b\} \rightarrow \{0, 1\}$, given by $p^x(\cdot) = p(x, \cdot)$ and $q^x(\cdot) = q(x, \cdot)$.

Step 2. Check if the pruning sequence is empty. If yes, there is only one vertex x left. If $\max\{p^x(b), q^x(b)\} = 1$, 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 $k \in \{0, 1, \dots, b\}$.

– 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 $k \in \{0, 1, \dots, b\}$.

– 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 $k \in \{0, 1, \dots, b\}$.

Go to *Step 2*.

The correctness of the algorithm can be easily proved by induction on n . We leave this routine proof to the reader. Clearly, the algorithm can be implemented so that it runs in time $O(b^2n + m)$.

1.4.1.8. Graphs of treewidth at most k

Graphs of treewidth at most k , also known as *partial k -trees*, generalize trees and 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 [ARNBORG 89]. It is easy to see that on trees, the EWIS problem admits a simple dynamic programming solution. With some care, the same approach can be generalized to graphs of bounded treewidth.

Let us first recall the definition of treewidth, and some related basic facts.

A *tree-decomposition* of a graph $G = (V, E)$ is a tree $T = (\mathcal{I}, F)$ where each vertex $i \in \mathcal{I}$ has a label $X_i \subseteq V$ such that :

$$(i) \cup_{i \in \mathcal{I}} X_i = V,$$

- (ii) For every edge $[u, v] \in E$, there exists an $i \in \mathcal{I}$ such that $u, v \in X_i$,
- (iii) For every $v \in V$, the vertices of T whose label contains v induce a connected subtree of T .

The *width* of such a decomposition is $\max_{i \in \mathcal{I}} |X_i|$. The *treewidth* of a graph G is the minimum k such that G has a tree-decomposition of width k .

- Any graph of treewidth k has a tree-decomposition $T = (\mathcal{I}, F)$ such that
- all the sets X_i in the decomposition have size $k + 1$,
 - if $[i, j] \in F$ then $|X_i \cap X_j| = k$.

Such a decomposition can be obtained in linear time from any tree-decomposition of G of width k . Also, given a graph of treewidth k , a tree-decomposition of G of width k can be obtained in linear time [BODLAENDER 96].

THÉORÈME 1.14.– *For every fixed k , the EWIS problem admits an $O(nb^2)$ algorithm for graphs of treewidth at most k .*

PREUVE.– Let $G = (V, E)$ be a weighted graph of treewidth k . First, we construct a special decomposition $T = (\mathcal{I}, F)$ of width k as mentioned above. 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 $p \in \{0, 1, \dots, b\}$, define the $\{0, 1\}$ -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 value of $\text{EWIS}(G, w, b)$ is *yes* if and only if there is a subset Z of the set X_r corresponding to the root r such that $\text{ewis}(r, Z, b) = 1$.

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

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

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 $X_i = X_j \setminus \{v\}$ for some vertex v . Also, $Y_i = Y_j$, since X_i does not add any new vertices. We can compute $\text{ewis}(i, Z, p)$ by the following formula

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

where \vee denotes the Boolean OR function.

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

From now on, assume that Z is independent. Recall that each of the sets X_{j_s} , for $s \in \{1, \dots, t\}$, 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 $s \in [t]$, let us denote by I_s that part of the set I which belongs to Y_{j_s} but not to Z , i.e., $I_s = I \cap (Y_{j_s} \setminus Z)$. 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, \dots, 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, $\text{ewis}(i, Z, p)$ will take the value 1 if and only if there exist nonnegative integers p_1, \dots, p_t such that :

- (i) $w(Z \cap X_{j_s}) \leq p_s \leq p$ for all $s \in [t]$,
- (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 $s \in [t]$.

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 $s \in [t]$.

On the other hand, the existence of such integers p_1, \dots, p_t implies that there are sets I'_1, \dots, I'_t such that, for all $s \in [t]$, 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 $I_s = I'_s \cap (Y_{j_s} \setminus Z)$. 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, \dots, I_t . Therefore,

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

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 $s \in [t]$.

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 $v \in I \setminus I'_1$ 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 $O(tb^2)$ 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, \dots, A_n and a positive integer b .

Question : Are there $a(i) \in A_i$ for all $i \in [n]$ such that $\sum_{i=1}^n a(i) = b$?

The correspondence is given by setting

$$* n = t,$$

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

and

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

A slight modification of the proof of Lemme 1.4 shows that this problem can be solved in time $O(tb^2)$.

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 EWIS problem for graphs of treewidth k is $O(nb^2)$.

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

1.4.1.9. 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 $\rho_{i \rightarrow j}$).

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 $\{1, \dots, k\}$ 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 [CORNEIL 05], 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 converse is generally not true, as the complete graphs show : for every $n \geq 2$, the clique-width of K_n is 2, while its treewidth is $n - 1$. 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 (see e.g. [COURCELLE 97]).⁷ The EWIS problem is no exception.

THÉORÈME 1.15.– *For every fixed k , the EWIS problem admits an $O(b^2l)$ algorithm for graphs of clique-width at most k , where l is the number of operations in a given k -expression for G .*

PREUVE.– Suppose that the labels are integers $\{1, \dots, k\} = [k]$. For every subset of labels $S \subseteq [k]$, let $\text{EWIS}(G, w, S, m)$ 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 determine the value of $\text{EWIS}(G, w, b)$ by first computing all the values for $\text{EWIS}(G, w, S, m)$, for every subset of labels $S \subseteq [k]$, and every $m \in [b]$. It is easy to see that this can be performed in time $O(b^2l)$ by the following dynamic programming algorithm.

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

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

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

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

This can be computed in time $O(b^2)$, similarly as in Corollaire 1.3.

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

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

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

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

⁷. 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 [OUM 05].

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

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

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 [STATON 98], as follows. A graph G is the line graph of a bipartite graph if and only if G is \mathcal{F} -free, where $\mathcal{F} = \{\text{claw}, \text{diamond}, C_5, C_7, \dots\}$. 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 Théorème 1.15.

COROLLAIRE 1.3.– The EWIS problem admits a pseudo-polynomial solution in each of the following graph classes :

- (*claw*, *co-claw*)-free graphs,
- (*gem*, *fork*, *co-P*)-free graphs (see Figure 1.4) and their subclass (*claw*, *diamond*, *co-P*)-free graphs,
- (P_5 , *diamond*)-free graphs.

PREUVE.– Each of the above subclasses is of bounded clique-width (see [BRANDSTÄDT 02, BRANDSTÄDT 03a, BRANDSTÄDT 04]).

Also, we can derive from Théorème 1.15 a particular complexity result for the exact perfect matching problem.

COROLLAIRE 1.4.– For every fixed k , the exact perfect matching problem admits a pseudo-polynomial algorithm for graphs of treewidth at most k .

PREUVE.— As shown by Gurski and Wanke [GURSKI 07], a set \mathcal{G} of graphs has bounded treewidth if and only if the set $L(\mathcal{G}) = \{L(G) : G \in \mathcal{G}\}$ has bounded clique-width. Since the exact perfect matching problem in \mathcal{G} is polynomially equivalent to the EWIS_α problem in the set $L(\mathcal{G})$, the statement follows from Théorème 1.15 and part (ii) of Lemme 1.1.

1.4.2. Modular decomposition

The idea of *modular decomposition* has been first described in the 1960s by Gallai [GALLAI 67], and also appeared in the literature under various other names such as *prime tree decomposition* [EHRENFEUCHT 90], *X-join decomposition* [HABIB 79], or *substitution decomposition* [MÖHRING 85]. 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 $\text{co-}G$ are both connected, then the maximal modules of G are pairwise disjoint. Moreover, from the above definition it follows that if U and U' are maximal modules, then either there are all possible edges between U and U' , 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 $\text{MODULAR_EWIS}(G, W, b)$ 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 weight). 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, b) , where $G = (V, E)$ is a graph, b is a positive integer and $W = (W_v : v \in V)$ with $W_v \subseteq [b]$ 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 $w : I \rightarrow [b]$ such that $w(v) \in W_v$ for all $v \in I$, and $\sum_{v \in I} w(v) = b$?

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, b) 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 $\text{EWIS}(G, W, b) = \text{yes}$ if and only if $\text{EWIS}(G', w', b) = \text{yes}$. However, working with the extended version enables us to apply modular decomposition to *arbitrary* graph classes.

Algorithm MODULAR_EWIS(G, W, b)

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

Output : $(\text{EWIS}(G, W, k) : k \in [b])$

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

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

and stop.

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

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

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

5. 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

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

6. For each $k \in [b]$, 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 $\{G, \text{co-}G\}$ is disconnected; moreover, if G and $\text{co-}G$ 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 $\mathcal{M}_1, \dots, \mathcal{M}_r$; 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.

THÉORÈME 1.16.— *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 the EWIS problem can be solved for graphs in \mathcal{G}^* in time $O(b^q n^p)$, then the EWIS problem can be solved for graphs in \mathcal{G} in time $O(b^q n^p + m)$.*

PREUVE.— 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 $O(n + m)$ [MCCONNELL 99]. 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 $G[\mathcal{M}_1], \dots, G[\mathcal{M}_r]$, where $\{\mathcal{M}_1, \dots, \mathcal{M}_r\}$ 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 $O(b^2 |V(G_U^0)|)$, since it is a generalized subset sum problem (cf. Lemme 1.4). If G_U is connected, then G_U^0 is a complete graph, and the problem can be solved trivially for G_U^0 in time $O(b |V(G_U^0)|)$.

If both G and $\text{co-}G$ 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 $O(b^q |V(G_U^0)|^p)$ 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 $O(b^q \sum_U |V(G_U^0)|^p)$. 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

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

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 the EWIS problem 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 1.4.

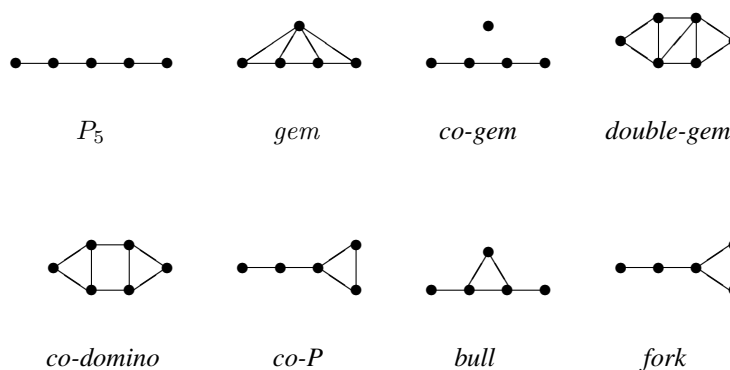


Figure 1.4. Some 5- and 6-vertex graphs

THÉORÈME 1.17.– *The EWIS problem is solvable in pseudo-polynomial time for each of the following classes :*

- $(P_5, \text{double-gem}, \text{co-domino})$ -free graphs (and their subclass, $(P_5, \text{co-P})$ -free graphs),
- $(\text{bull}, \text{fork})$ -free graphs,
- $(\text{co-P}, \text{fork})$ -free graphs,
- (P_5, fork) -free graphs.

PREUVE.– This theorem essentially follows from Théorème 1.16 and the results in [BRANDSTÄDT 04a] and [HOÀNG 89] (see also [BRANDSTÄDT 04b] for some applications of modular decomposition to the WIS problem). We briefly summarize the main ideas.

Every prime $(P_5, \text{double-gem}, \text{co-domino})$ -free graph is $2K_2$ -free (the complementary version of this statement is proved in [HOÀNG 89]). Since we can easily extend Théorème 1.4 to the extended version of the EWIS problem, this implies the result for $(P_5, \text{double-gem}, \text{co-domino})$ -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 $O(nb)$ subproblems in P_4 -free graphs, which can be done by modular decomposition. It is well known (see e.g. [CORNEIL 81]) 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 [BRANDSTÄDT 04a], it is shown that prime graphs that contain a *co-gem* and are either *(bull, fork)*-free, *(co-P, fork)*-free or *(P₅, 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 Théorème 1.16, this concludes the proof.

1.5. 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 the EWIS problem 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 the EWIS problem is known, and whose sets of forbidden induced subgraphs differ only in two graphs.

Recall that the line graphs of bipartite graphs are precisely the *(claw, diamond, C₅, C₇, ...)*-free graphs. Replacing the diamond in the above characterization by its subgraph C_3 results in a smaller class of *(K_{1,3}, C₃, C₅, C₇, ...)*-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 Corollaire 1.3 and Théorème 1.14, 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 $L(Bip)$, we obtain a class of graphs that strictly contains line graphs of bipartite graphs. This class of *(K_{1,4}, diamond, C₅, C₇, ...)*-free graphs contains the class of *(K_{1,4}, C₃, C₅, C₇, ...)*-free graphs, which is precisely the class of bipartite graphs of maximum degree at most 3. The results of Section 1.3.1 imply that the problem is strongly **NP**-complete for this class, and hence also for the larger class of *(K_{1,4}, diamond, C₅, C₇, ...)*-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 the EWIS problem is known, as the following diagram shows.

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

Acknowledgments. Part of this research was carried out while the first author was visiting the second author at LAMSADE at the University of Paris 9 supported by DIMACS. The support of this institution is gratefully acknowledged.

1.6. Bibliographie

- [ALEKSEEV 91] ALEKSEEV V.E., « On the number of maximal independent sets in graphs from hereditary classes », *Combinatorial-algebraic methods in discrete optimization*, University of Nizhny Novgorod, p. 5-8, 1991 (in Russian).
- [ARNBORG 89] ARNBORG S., PROSKUROWSKI A., « Linear time algorithms for NP-hard problems restricted to partial k -trees », *Discrete Appl. Math.*, vol. 23, p. 11-24, 1989.
- [BALAS 89] BALAS E., YU C.S., « On graphs with polynomially solvable maximum-weight clique problem », *Networks*, vol. 19, p. 247-253, 1989.
- [BANDELT 86] BANDELT H.J., MULDER H.M., « Distance-hereditary graphs », *J. Combin. Theory Ser. B*, vol. 41, p. 182-208, 1986.
- [BARAHONA 87] BARAHONA F., PULLEYBLANK W.R., « Exact arborescences, matchings, and cycles », *Discrete Appl. Math.*, vol. 16, p. 91-99, 1987.
- [BERGE 73] BERGE C., *Graphs and hypergraphs*, North Holland, Amsterdam, 1973.
- [BŁAŻEWICZ 06] BŁAŻEWICZ J., FORMANOWICZ P., KASPRZAK M., SCHUURMAN P., WOEGINGER G., « A polynomial time equivalence between DNA sequencing and the exact perfect matching problem », *Discrete Optim.*, (2006), doi :10.1016/j.disopt.2006.07.004.
- [BODLAENDER 96] BODLAENDER H.L., KLOKS T., « Efficient and constructive algorithms for the pathwidth and treewidth of graphs », *J. Algorithms*, vol. 21, p. 358-402, 1996.
- [BRANDSTÄDT 99] BRANDSTÄDT A., LE V.B., SPINRAD J., *Graph classes : a survey.*, SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 1999.
- [BRANDSTÄDT 02] BRANDSTÄDT A., MAHFUD S., « Maximum weight stable set on graphs without claw and co-claw (and similar graph classes) can be solved in linear time », *Inform. Process. Lett.*, vol. 84, p. 251-259, 2002.
- [BRANDSTÄDT 03] BRANDSTÄDT A., DRAGAN F.F., « On linear and circular structure of (claw, net)-free graphs », *Discrete Appl. Math.*, vol. 129, p. 285-303, 2003.

- [BRANDSTÄDT 03a] BRANDSTÄDT A., LE H.-O., VANHERPE J.-M., « Structure and stability number of chair-, co-P- and gem-free graphs revisited », *Inform. Process. Lett.*, vol. 86, p. 161-167, 2003.
- [BRANDSTÄDT 04] BRANDSTÄDT A., « $(P_5, \text{diamond})$ -free graphs revisited : structure and linear time optimization », *Discrete Appl. Math.*, vol. 138, p. 13-27, 2004.
- [BRANDSTÄDT 04a] BRANDSTÄDT A., LE V.B., DE RIDDER H.N., « Efficient robust algorithms for the maximum weight stable set problem in chair-free graph classes », *Inform. Process. Lett.*, vol. 89, p. 165-173, 2004.
- [BRANDSTÄDT 04b] BRANDSTÄDT A., HOÀNG C.T., VANHERPE J.-M., « On minimal prime extensions of a four-vertex graph in a prime graph », *Discrete Math.*, vol. 288, p. 9-17, 2004.
- [BROERSMA 99] BROERSMA H., KLOKS T., KRATSCHE D., MÜLLER H., « Independent sets in asteroidal triple-free graphs », *SIAM J. Discrete Math.*, vol. 12, p. 276-287, 1999.
- [COGIS 05] COGIS O., THIERRY E., « Computing maximum stable sets for distance-hereditary graphs », *Discrete Optim.*, vol. 2, p. 185-188, 2005.
- [CORNEIL 81] CORNEIL D.G., LERCHS H., STEWART-BURLINGHAM L., « Complement reducible graphs », *Discrete Appl. Math.*, vol. 3, p. 163-174, 1981.
- [CORNEIL 05] CORNEIL D.G., ROTICS U., « On the relationship between clique-width and treewidth », *SIAM J. Comput.*, vol. 34, p. 825-847, 2005.
- [COURCELLE 97] COURCELLE B., « The expression of graph properties and graph transformations in monadic second-order logic », *Handbook of graph grammars and computing by graph transformation*, vol. 1, p. 313-400, World Sci. Publishing, River Edge, NJ, 1997.
- [DAMIAND 01] DAMIAND G., HABIB M., PAUL C., « A simple paradigm for graph recognition : application to cographs and distance hereditary graphs », *Combinatorics and computer science (Palaiseau, 1997). Theoret. Comput. Sci.*, vol. 263, p. 99-111, 2001.
- [DAWANDE 01] DAWANDE M., KESKINOCAK P., SWAMINATHAN J.M., TAYUR S., « On Bipartite and Multipartite Clique Problems », *J. Algorithms*, vol. 41, p. 388-403, 2001.
- [DEĚNEKO 06] DEĚNEKO V.G., WOEGINGER G.J., « On the robust assignment problem under a fixed number of cost scenarios », *Oper. Res. Lett.*, vol. 34, p. 175-179, 2006.
- [EHRENFEUCHT 90] EHRENFEUCHT A., ROZENBERG G., « Primitivity is hereditary for 2-structures », *Theoret. Comput. Sci.*, vol. 70, p. 343-358, 1990.
- [FRANK 76] FRANK A., « Some polynomial algorithms for certain graphs and hypergraphs », *Proc. of the 5th Brit. Comb. Conf., Aberdeen 1975, Congr. Numer. XV*, p. 211-226, 1976.
- [GALLAI 67] GALLAI T., « Transitiv orientierbare graphen », *Acta Math. Acad. Sci. Hungar.*, vol. 18, p. 25-66, 1967.
- [GAREY 79] GAREY M. R., JOHNSON D. S., *Computers and intractability. A guide to the theory of NP-completeness*, W. H. Freeman, San Francisco, 1979.
- [GRÖTSCHHEL 84] GRÖTSCHHEL M., LOVÁSZ L., SCHRIJVER A., « Polynomial algorithms for perfect graphs », *Topics on perfect graphs*, p. 325-356, North-Holland Math. Stud., 88, North-Holland, Amsterdam, 1984.

- [GURSKI 07] GURSKI F., WANKE E., « Line graphs of bounded clique-width », *Discrete Math.*, to appear, 2007.
- [HABIB 79] HABIB M., MAURER M.C., « On the X-join decomposition for undirected graphs », *Discrete Appl. Math.*, vol. 1, p. 201-207, 1979.
- [HOÀNG 89] HOÀNG C.T., REED B., « Some classes of perfectly orderable graphs », *J. Graph Theory*, vol. 13, p. 445-463, 1989.
- [HSU 99] HSU W.-L., MA T.-H., « Fast and simple algorithms for recognizing chordal comparability graphs and interval graphs », *SIAM J. Comput.*, vol. 28, p. 1004-1020, 1999.
- [KARZANOV 87] KARZANOV A.V., « Maximum matching of given weight in complete and complete bipartite graphs », *Cybernetics*, vol. 23, p. 8-13, 1987. Translation from *Kibernetika*, vol. 1, p. 7-11, 1987.
- [LECLERC 86] LECLERC M., « Polynomial time algorithms for exact matching problems », *Master's Thesis, University of Waterloo, Waterloo, 1986*.
- [MANNINO 07] MANNINO C., ORIOLO G., RICCI F., CHANDRAN S., « The stable set problem and the thinness of a graph », *Oper. Res. Lett.*, vol. 35, p. 1-9, 2007.
- [MCCONNELL 99] MCCONNELL R.M., SPINRAD J.P., « Modular decomposition and transitive orientation », *Discrete Math.*, vol. 201, p. 199-241, 1999.
- [MÖHRING 85] MÖHRING R.H., « Algorithmic aspects of comparability graphs and interval graphs », In : *I. Rival (Ed.), Graphs and Orders, D. Reidel, Boston, 1985*. p. 41-101.
- [MULMULEY 87] MULMULEY K., VAZIRANI U., VAZIRANI V.V., « Matching is as easy as matrix inversion », *Combinatorica*, vol. 7, p. 105-113, 1987.
- [MURPHY 92] MURPHY O.J., « Computing independent sets in graphs with large girth », *Discrete Appl. Math.*, vol. 35, p. 167-170, 1992.
- [OKAMOTO 05] OKAMOTO Y., UNO T., UEHARA R., « Linear-time counting algorithms for independent sets in chordal graphs », *Graph-theoretic concepts in computer science, Lecture Notes in Comput. Sci., 3787, Springer, Berlin, 2005*, p. 433-444.
- [OUM 05] OUM S.I., « Approximating rank-width and clique-width quickly », *Graph-theoretic concepts in computer science, Lecture Notes in Comput. Sci., 3787, Springer, Berlin, 2005*, p. 49-58.
- [PAPADIMITRIOU 82] PAPADIMITRIOU C.H., YANNAKAKIS M., « The complexity of restricted spanning tree problems », *J. ACM*, vol. 29, p. 285-309, 1982.
- [POLJAK 74] POLJAK S., « A note on stable sets and coloring of graphs », *Comment. Math. Univ. Carolinae*, vol. 15, p. 307-309, 1974.
- [PRISNER 95] PRISNER E., « Graphs with few cliques », *Inform. Process. Lett.*, vol. 1, 2, (Kalamazoo, MI, 1992), p. 945-956, 1995, Wiley-Interscience Publ., Wiley, New York.
- [RAMALINGAM 88] RAMALINGAM J., PANDU RANGAN C., « A unified approach to domination problems on interval graphs », *Inform. Process. Lett.*, vol. 27, p. 271-274, 1988.
- [STATON 98] STATON W., WINGARD G.C., « On line graphs of bipartite graphs », *Util. Math.*, vol. 53, p. 183-187, 1998.

- [SUPOWIT 87] SUPOWIT K.J., « Finding a maximum planar subset of a set of nets in a channel », *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-6, p. 93-94, 1987.
- [TARJAN 85] TARJAN R.E., « Decomposition by clique separators », *Discrete Math.*, vol. 55, p. 221-232, 1985.
- [THOMAS 06] THOMAS R., « A survey of Pfaffian orientations of graphs », *Proc. of the Internat. Congress of Mathematicians, Vol. III (Madrid, 2006)*, p. 963–984, EMS Publishing House, 2006.
- [TSUKIYAMA 77] TSUKIYAMA S., IDE M., ARIYOSHI H., SHIRAKAWA I., « A new algorithm for generating all the maximal independent sets », *SIAM J. Comput.*, vol. 6, p. 505-517, 1977.

Chapitre 2

Index

- Balanced biclique problem, 8
- Clique, 8
- Clique tree, 18
- Clique-width, 3, 29, 31
- Clique-width expression, 29
- Dynamic programming algorithm, 13, 15–17, 21, 23, 25, 30
- Exact arborescence problem, 1
- Exact cut problem, 1
- Exact matching problem, 2, 36
- Exact perfect matching problem, 1–4, 7, 31
- Exact spanning tree problem, 1
- Exact weighted independent set problem (EWIS), 2, 3, 5, 7, 14, 32, 36
- Exact weighted maximum independent set problem (EWIS _{α}), 2, 3, 5, 8, 12
- Graphs, 4
 - AT-free, 3, 21
 - Bipartite, 2, 3, 7, 8
 - (*bull, fork*)-free, 35
 - Chordal, 3, 7, 18
 - Circle, 3, 17, 23
 - (*claw, net*)-free, 3, 22
 - (*co-P, fork*)-free, 35
 - Co-comparability, 20
 - Distance-hereditary, 3, 23
 - Forests, 7
 - Interval, 3, 15
 - k*-thin, 3, 16
 - Line graph, 2, 3, 7
 - Line graphs of bipartite, 4, 7, 31, 36, 37
 - mK_2 -free, 3, 14
 - (P_5 , *co-P*)-free, 35
 - (P_5 , *double-gem, co-domino*)-free, 35
 - Perfect, 11, 20
 - Permutation, 20
 - (P_5 , *fork*)-free, 35
 - prime, 32
- Independent set, 2
 - Maximum independent set ($\alpha(G)$), 2, 10
- Induced subgraph, 4, 11, 23
- Matching, 1
 - Perfect matching, 1
- Modular decomposition, 32, 33
- Pruning sequence, 23
- Pseudo-polynomial time, 1, 3, 6, 14, 15, 18, 22, 26, 30, 31, 35, 36
- Stable set, 2
- Strong NP-completeness, 3, 8, 12
- Tree-decomposition, 26
- Treewidth, 3, 8, 26, 29
- Weighted independent set problem, 2, 5, 14