The Independent Set Problem and the Thinness of a Graph

Sunil Chandran (sunil@mpi-sb.mpg.de)
Carlo Mannino (mannino@dis.uniroma1.it)
Gianpaolo Oriolo (oriolo@disp.uniroma2.it)

Abstract

Finding an optimal independent set on an interval graph is easy and weighted independent set problems on superclasses of interval graphs frequently arise in modelling practical applications. Motivated by a relevant one, arising in wireless telecommunication, we introduce a new superclass of interval graphs that, unlike most others, allows to carry out the optimization task in an effective way. This superclass is related to the definition of a new graph invariant, called 'thinness", which appears to be strongly related to other well-known graph invariants: the pathwidth and the boxicity. More in detail, we do the following:

We show that a maximum weighted independent set on a graph $G(V, E)$ with thinness k may be found, when a certain representation is at hand, in $O(|V(G)| \cdot (\rho + 1)^{k-1})$-time, with ρ bounded by $\Delta(G)$. A graph has thinness 1 if and only if it is an interval graph, and in this case there exists a simple representation with $\rho = 0$ (and viceversa).

We investigate some theoretical properties of the thinness. First, we relate the thinness to the pathwidth of a graph. We show that, if q is the pathwidth of G, then the thinness of G is at most $q + 1$ and there exists a suitable representation for which $\rho \leq 1$: this implies a well-known result on the complexity of computing an optimal independent set in a graph for which a path decomposition is given. We point out that the gap between pathwidth and thinness may very high since a clique with n vertices has pathwidth $n - 1$ but is interval.

Then we consider a different generalization of interval graphs, related to the definition of boxicity of a graph. We show that the boxicity is a lower bound to the thinness but also that the $r \times r$ grid graph has boxicity two and thinness $O(n)$. Finally, we provide with a couple of general upper bounds on the thinness of a graph.

We then turn to applications. In fact, weighted independent set problems on graphs with small thinness naturally arise in the solution of a major optimization problem in telecommunication systems, namely the frequency assignment problem (FAP); but also, for example, in formulating and solving the single machine scheduling problem. We show that an efficient search in exponential neighborhoods for the FAP maybe done in polynomial time by a dynamic programming algorithm. This led us to improving the best known solutions for several benchmark instances of the COST259 test-bed [9], a set of large real life instances of GSM networks operating in Germany.

Keywords: Interval Graphs, Independent Sets, Pathwidth, Boxicity, Frequency Assignment.

1 Introduction

Interval graphs were introduced by Hajos [15] in 1957 and since then have been widely studied. First, there are many applications, among them scheduling and computational biology, which can be modelled as optimization problems on suitable interval graphs. Moreover, interval graphs can be recognized in linear time and many classical optimization problems, e.g. the maximum independent set problem, are easy for them. Good sources for this theory are [10] and [13].

There exist several characterizations of interval graphs; the most popular one says that a graph $G(V, E)$, with vertex set $V = \{v_1, \ldots, v_n\}$ is an *interval graph* if and only if there exists a set of intervals of the real line $\mathcal{M} = \{I_1, \ldots, I_n\}$ such that $v_i v_j \in E$ if and only if $I_i \cap I_j \neq \emptyset$.

Several generalizations of interval graphs have been proposed. Some of these generalizations "naturally" extend the basic definition, e.g. circular arc graphs and intersection graphs of rectangles. In some cases these generalizations are useful for modelling specific applications, for instance, independent set problems in intersection graphs of rectangles occur in the automatic label placement [11] and fleet maintenance [21] problems ($S \subset V$ is a *independent set* of G if and only if $uv \notin E$ for all $u, v \in S$). Unfortunately, recognizing if a graph can be obtained as the intersection graphs of a set of rectangles is NP-hard and, even worse, the maximum independent set problem, is hard, even if the "rectangle representation" is given [12, 16].

Motivated by a relevant application arising in wireless telecommunication, we introduce a superclass of interval graphs for which the maximum (weighted) independent set problem may be solved by dynamic programming in polynomial time when a certain representation is given.

The new class is introduced by extending a classical characterization [20, 23]: a graph $G(V, E)$ is interval if and only if there exists an ordering $\{v_1, \ldots, v_n\}$ of the vertices such that, for each triple (r, s, t) such that $r < s < t$, if $v_t v_r \in E$, then $v_t v_s \in E$: if such ordering is at hand, an $O(|V|)$-time dynamic programming algorithm finds a maximum weighted independent set [14].

We generalize this property. Namely, we define a graph $G(V, E)$ to be *k-thin* if there exist an ordering $\{v_1, \ldots, v_n\}$ of V and a *partition* of V into k classes such that, for each triple (r, s, t) such that $r < s < t$, if v_r, v_s belong to the same class and $v_t v_r \in E$, then $v_t v_s \in E$.

Therefore, a graph is interval if and only if it is 1-thin. If we are given for a (k-thin) graph $G(V, E)$ such an ordering and a partition, then a maximum weighted independent set S may be found via dynamic programming in $O(|V| \cdot (\rho + 1)^{k-1})$-time (Theorem 3.7), where ρ is always bounded by the maximum degree of a vertex. Also, since for interval graphs, $\rho = 0$ this extends the result in [14]. These results can be extended to the case (relevant in our applications) where S has to satisfy some side constraints (Theorem 3.9).

We define the *thinness* of a graph G ($thin(G)$) as the minimum k for which G is k-thin. We relate the thinness to the pathwidth of a graph and show that the thinness is bounded by the pathwidth +1 (Theorem 4.5). The theorem also gives a simple procedure for building, from a path decomposition of a graph, a consistent ordering and a partition such that $\rho \leq 1$: together with Theorem 3.7 this implies the well-known result that a maximum weighted independent set on a graph may be found in $O(|V(G)| \cdot 2^q)$-time if a path decomposition of width q is given.

Then we consider a different generalization of interval graphs, related to the definition of boxicity of a graph. For a graph G the boxicity $b(G)$ is the minimum dimension d such that G is the intersection graph of boxes (parallelepiped) in d-dimensional space. In particular, a graph has boxicity two if and only if it is the intersection graph of rectangles. We show that the boxicity is a lower bound to the thinness but also that the $r \times r$ grid graph has boxicity two and thinness $O(n)$. Finally, we provide with a couple of general upper bounds on the thinness of a graph.

The application which inspired our generalization is discussed in Section 5.1: namely, the frequency assignment problem (FAP) in GSM networks, that is the problem of assigning transmission frequencies to transmitters of a wireless network, so as to minimize the overall interference level, a crucial issue for increasing the quality of service. For suitable subsets T of transmitters, this problem can be formulated as a maximum weighted independent set problem on a graph G_T; since an ordering and a partition showing that G_T is $|T|$-thin can be easily built (Theorem 5.1), the problem can be then efficiently solved by our dynamic program (usually, $\rho \leq 2$) . This allows us to implement an effective *search in exponential neighborhoods*, as defined by Deineko and Woeginger in [8]: indeed, by encapsulating our methodology into a standard simulated annealing procedure we improved the best known solutions for most instances of the COST259 test bed [9], which consists of a number of large real-life instances of the FAP in GSM networks.

We eventually outline another application where independent set problems on k-thin graphs "naturally" arise: the single machine scheduling problem.

In the remaining part of this introduction we recall some graph theoretical notations and definitions that are used throughout the paper. A graph will always be simple and undirected. We denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G. If W is a subset $W \subseteq V$, we denote by $G[W]$ the subgraph of G induced by W. We simply write $G - W$ for $G[V - W]$; when $W = \{v\}$, we write $G - v$. For each $v \in V$ we denote by $\delta(v)$ the *degree* of v. For each $v \in V$ we denote by $w(v) \in \mathcal{R}_+$ the *weight* of v; for any $S \subset V$, the *weight* of S is the quantity $w(S) = \sum_{u \in S} w(u)$. We denote by $\alpha(G)$ and by $\alpha_w(G)$ the maximum size and weight of an independent set of G, respectively. We denote by $\Delta(G)$ the maximum degree of a vertex of G.

For a given family $\mathcal{M}$ of sets, the *intersection graph* $G_{\mathcal{M}}$ has $\mathcal{M}$ as vertex set and two vertices are adjacent if the corresponding sets have non-empty intersection.

Definition 1.1 *Let $G(V, E)$ be a graph. For each $v \in V$ and $X \subset V$ we denote by:*

- *$N(v)$ the* neighborhood *of v, that is the set of vertices adjacent to v*
- *$\overline{N}(v)$ the set of vertices different from v and not adjacent to v, i.e. $\overline{N}(v) = V - (N(v) \cup \{v\})$*
- *$N(X)$ the set of neighbors of X, i.e. $N(X) = \{u \in V - X : u \in N(v)$ for some $v \in X\}$*

We often deal with partitions and ordering of the set V. For our purposes, a *k-partition* of V is just a partition of V into k classes $V^1, \ldots, V^k$. All our *orderings* are *linear* (i.e. not partial), but for shortness, we refer to them just as orderings. We usually represent an ordering by writing V as $\{v_1, v_2, \ldots, v_n\}$, meaning that $v_j > v_i$ if and only if $j > i$.

Definition 1.2 *Let $G(V, E)$ be a graph with an ordering $\{v_1, \ldots, v_n\}$ and a k-partition $(V^1, \ldots, V^k)$ of V. For each $1 \leq j \leq n$:*

- *$\overline{N}(v_j)_<$ is the set of vertices of V which are lower than v_j and non-adjacent to v_j, i.e. $\overline{N}(v_j)_< = \{v_i \in V : i < j$ and $v_i v_j \notin E\}$.*
- *For each $1 \leq h \leq k$, $\overline{N}(v_j, h)_<$ is the set of vertices of V^h which are lower than v_j and non-adjacent to v_j, i.e. $\overline{N}(v_j, h)_< = \{v_i \in V^h : i < j$ and $v_i v_j \notin E\} = V^h \cap \overline{N}(v_j)_<$.*

Observe that, by definition, $\overline{N}(v_n)_< = \overline{N}(v_n)$ and $\overline{N}(v_j)_< = \bigcup_h \overline{N}(v_j, h)_<$.

2 Independent set problems on interval graphs

A graph is an *interval graph* if it is the intersection graph of a set of intervals of the real line. Let $\mathcal{M} = \{I_1, \ldots, I_n\}$ be a set of intervals, and let $G_{\mathcal{M}}$ be the corresponding intersection graph: $G_{\mathcal{M}}$ is an interval graph and $\mathcal{M}$ is called the *interval representation* of $G_{\mathcal{M}}$.

An interval graph $G(V, E)$ may be recognized, and an interval representation provided, in $O(|V|+|E|)$-time [4]. If we are given an interval representation of G, then a maximum weighted independent set can be found in $O(|V| \log |V|)$-time via a dynamic programming algorithm: this was first observed in [14], and it is briefly described in the following - the main ingredients will be extended to our generalization in Section 3.

Let $G = (V, E)$ be a (general) graph with weights $w(v)$ for each vertex $v \in V$. For each $X \subseteq V$ denote by $\alpha_w(X)$ the maximum weight of an independent set in $G[X]$. Let $v \in V$ be any vertex of G and let S be a maximum weighted independent set of G. If $v \notin S$ then also S is a maximum weighted independent set in $G[V - v]$; if $v \in S$ then $S - v$ is a maximum weighted independent set in $G[V - (v \cup N(v))] = G[\overline{N}(v)]$ (see Definition 1.1). So, the following recursion correctly evaluates $\alpha_w(V)$:

$$\alpha_w(V) = \max \begin{cases} \alpha_w(V - v) \\ \alpha_w(\overline{N}(v)) + w(v) \end{cases} \tag{1}$$

The above recursion can be solved by observing that both terms involve graphs smaller than G: when the current graph (in the tree generated by the recursion) is sufficiently small, we may resort to some type of "direct" solution method. A major drawback of recursion (1) is that the recursive tree may grow very large, in general exponentially with $|V|$. However, if G is an interval graph, one can keep the size of the tree bounded by $|V|$.

Our first "ingredient" is the following theorem providing a characterization of interval graphs.

Theorem 2.1 [20, 23] *A graph $G(V, E)$ is an interval graph if and only if there exists an ordering $\{v_1, v_2, \ldots, v_n\}$ of V such that, for each triple (i, j, k), with $1 \le i < j < k \le n$, if $v_k v_i \in E$, then $v_k v_j \in E$.*

In order to construct such an ordering, it is enough to order intervals (vertices) by increasing right endpoints (and of course this can be done in $O(|V| \log |V|)$-time).

Our second ingredient is the next lemma, the proof is trivial, showing that an ordering satisfying Theorem 2.1 has a nice property called *consistency* (see Definition 1.2 for $\overline{N}(v_j)_<$).

Lemma 2.2 *Let $G(V, E)$ be a graph and $\{v_1, \ldots, v_n\}$ an ordering of V. The following statements are equivalent:*

(i) for each triple (i, j, k), with $1 \le i < j < k \le n$, if $v_k v_i \in E(G)$ then $v_k v_j \in E(G)$;

(ii) [Consistency] *for each $1 \le j \le n$, $\overline{N}(v_j)_< = \{v_1, v_2, \ldots, v_{|\overline{N}(v_j)_<|}\}$.*

In other words, if an ordering $\{v_1, \ldots, v_n\}$ is consistent, then, for each j, the only vertices of $\{v_1, \ldots, v_{j-1}\}$ adjacent to v_j are the last $(j-1) - |\overline{N}(v_j)_<|$ ones. The following corollary is a straight consequence of Theorem 2.1 and Lemma 2.2.

Corollary 2.3 *The following statements are equivalent:*

(i) G is interval;

(ii) There exists a ordering $\{v_1, \dots v_n\}$ of $V(G)$ which is consistent.

Let $G = (V, E)$ be a (interval) graph with a consistent ordering $\{v_1, \dots v_n\}$ of $V(G)$ and weights $w(v_j)$ for each vertex v_j. Let $\alpha_w(0) = 0$ and, for each $1 \le j \le n$, denote by $\alpha_w(j)$ the maximum weight of an independent set $S_j \subseteq \{v_1, \dots, v_j\}$. If $v_j \notin S_j$ then also S_j is a maximum weighted independent set in $\{v_1, \dots, v_{j-1}\}$. If $v_j \in S_j$ then $S_j - v_j$ is a maximum weighted independent set in $G[\{v_1, \dots, v_{j-1}\} - N(v_j)] = G[\overline{N}(v_j)_<]$ by Definition 1.2. Since the ordering is consistent, by Lemma 2.2 $\overline{N}(v_j)_< = \{v_1, v_2, \dots, v_{|\overline{N}(v_j)_<|}\}$. So, the following recursion correctly evaluates $\alpha_w(j)$:

$$\alpha_w(j) = \max \begin{cases} \alpha_w(j-1) \\ \alpha_w(|\overline{N}(v_j)_<|) + w(v_j) \end{cases} \tag{2}$$

and $\alpha_w(n) = \alpha_w(G)$ may be found in $O(n)$-time by sequentially computing $\alpha_w(1), \alpha_w(2), \dots \alpha_w(n)$.

3 Independent set problems on a superclass of interval graphs

The existence of a consistent ordering is at the basis of the dynamic program (2). On the other hand, there is such an ordering if and only if a graph is interval (Corollary 2.3). In this section, however, we deal with a different consistency property, involving also a *partition* of the vertex set, which does hold for every graph: this property can be exploited to define an efficient algorithm for the maximum weighted independent set problem for every graph for which the ordering and the partition are given.

As usual, let $G = (V, E)$ be a graph with an ordering $\{v_1, \dots v_n\}$ of V. Suppose that also we are given a *k-partition* $V^1, \dots, V^k$ of V. Denote by p^h the size of each class V^h and assume that $V^h = \{v_1^h, \dots, v_{p^h}^h\}$, with $v_1^h < v_2^h < \dots < v_{p^h}^h$ (i.e. with respect to the ordering).

The following lemma is a natural extension of Lemma 2.2. Again, we omit the simple proof (see Definition 1.2 for $\overline{N}(v_j, h)_<$).

Lemma 3.1 *Let $G(V, E)$ be a graph with a k-partition $(V^1, \dots, V^k)$ and an ordering $\{v_1, v_2, \dots, v_n\}$ of V. The following statements are equivalent:*

(i) for each triple (i, j, k) such that $1 \le i < j < k \le n$, if $v_k v_i \in E$ and v_i and v_j belong to the same class, then $v_k v_j \in E$;

(ii) [Consistency] *for each vertex v_j and each class V^h, $\overline{N}(v_j, h)_< = \{v_1^h, v_2^h, \dots, v^h_{|\overline{N}(v_j,h)_<|}\}$.*

In other words, if an ordering and a k-partition are consistent, then, for each vertex v_j and each class V^h, the only vertices of V^h which are *lower* than v_j and *non-adjacent* to v_j are the first $|\overline{N}(v_j, h)_<|$ vertices of V^h. An example is given in Fig. 1, where the (ordered) set of vertices is $V = \{v_1, v_2, \dots, v_{13}\}$, and the consistent 2-partition is $\mathcal{V} = (V^1, V^2)$ with $V^1 = \{v_3, v_4, v_5, v_6, v_8, v_{11}, v_{12}\}$ and $V^2 = \{v_1, v_2, v_7, v_9, v_{10}, v_{13}\}$.

Now, let $\alpha_w(j^1, \dots, j^k)$ be the maximum weight of an independent set $S \subseteq \bigcup_{h=1..k}\{v_1^h, \dots, v_{j^h}^h\}$ and $\mathcal{K}$ the set of k-tuples $(j^1, \dots, j^k) : 0 \le j^h \le p^h$. Without loss of generality, let $v_{j^1}^1 > v_{j^h}^h$

$\forall\ h \neq 1$. As we show in the theorem, the following dynamic program correctly evaluates $\alpha_w(j^1, ..., j^k)$:

$$\alpha_w(j^1, ..., j^k) = \max \begin{cases} \alpha_w(j^1 - 1, j^2, j^3, \dots j^k) \\ w(u) + \alpha_w(b^1, \dots, b^k) \end{cases} \tag{3}$$

where both $(b^1, \dots, b^k)$ and $(j^1 - 1, j^2, j^3, \dots j^k)$ are strictly dominated by $(j^1, \dots, j^k)$. Hence, in order to compute $\alpha_w(G) = \alpha_w(p^1, \dots, p^k)$, it is enough to (orderly) compute α_w only for the k-tuples of $\mathcal{K}$, whose number is bounded by $(\frac{|V|}{k} + 1)^k$.

Theorem 3.2 *If for a graph G we are given an ordering and a k-partition which are consistent, then a maximum weighted independent set of G may be found in $O(\frac{|V|}{k})^k$-time.*

3.1 A strengthening of Theorem 3.2

The complexity established by Theorem 3.2 can be refined by observing that indeed only a small subset of k-tuples of $\mathcal{K}$ will be actually enumerated. By simple counting arguments we give an upper bound on the maximum size of this subset. As usual, we are given a graph $G = (V, E)$ with an ordering and a k-partition of V which are consistent, denoted for our convenience as $\sigma = \{v_1, \dots v_n\}$ and $\mathcal{V} = (V^1, \dots, V^k)$, respectively.

For each $j \in \{1, \dots, n\}$, denote as $p(j)$ the highest *index* $i < j$ such that $v_i v_j \notin E$. Recall from Definition 1.2 that $\overline{N}(v_j)_<$ is the set of vertices of V which are smaller than v_j and non-adjacent to v_j. It follows that $v_{p(j)}$ is the highest vertex in $\overline{N}(v_j)_<$ and $p(j) \geq |\overline{N}(v_j)_<|$. In the example of Fig. 1, we have, for instance, that $p(12) = 9 > |\overline{N}(v_{12})_<| = 8$ and $p(10) = 8 > |\overline{N}(v_{10})_<| = 6$.

Proposition 3.3 *If $p(j) = |\overline{N}(v_j)_<|$ for every j then G is an interval graph.*

Proof. Assume that $\overline{N}(v_j)_< = \{v_{j_1}, \dots, v_{j_{|\overline{N}(v_j)_<|}}\}$. Since $|\overline{N}(v_j)_<| = p(j)$, then we must have $j_1 = 1, j_2 = 2, \dots, j_{|\overline{N}(v_j)_<|} = p(j)$ and so for each $v_j \in V$, we have that $\overline{N}(v_j)_< = \{v_1, v_2, \dots, v_{p(j)}\}$. So the ordering is consistent since it satisfies property (*ii*) of Lemma 2.2. □

For each $v_j \in V$ and each class V^h, denote as $\rho(v_j, h)$ the number of vertices of V^h that are smaller than $v_{p(j)}$ and are adjacent to v_j.

Proposition 3.4 $p(j) - |\overline{N}(v_j)_<| = \sum_h \rho(v_j, h)$.

Proof. Let $u = v_{p(j)}$. Let $q(u, h)$ be the number of vertices in each class not larger than u, i.e. $q(u, h) = |V^h \cap \{v_1, \dots, v_{p(j)}\}|$. It follows that $p(j) = \sum_h q(u, h)$. Also, by definition, $\rho(v_j, h) = q(u, h) - |\overline{N}(v_j, h)_<|$, for each h. Finally observe that $p(j) - |\overline{N}(v_j)_<| = p(j) - \sum_h |\overline{N}(v_j, h)_<| = \sum_h q(u, h) - \sum_h |\overline{N}(v_j, h)_<| = \sum_h \rho(v_j, h)$. □

Finally, let $\rho = \rho(\sigma, \mathcal{V}) = \max_{j,h} \rho(v_j, h)$[1].

We have now two simple corollaries: the former follows from Propositions 3.3 and 3.4; the latter from the definition of ρ and since, for any vertex v and each class V^h, $\rho(v, h) \leq \delta(v)$.

[1]Even if the value of ρ, as well as the values of $\rho(v_j, h)$, depends on the (consistent) ordering and k-partition we are given, when there is no risk of confusion, we simply write ρ. Moreover, for a given pair $(\sigma, \mathcal{V})$, ρ can be easily evaluated.

Corollary 3.5 *If $\rho = 0$ then G is an interval graph (and viceversa if G is an interval graph then $\rho = 0$ for any ordering satisfying Lemma 2.2).*

Proposition 3.6 $\rho \leq \Delta(G)$.

If we consider again the example of Fig. 1, we have $\rho(12, 1) = 1$ – recall that $p(12) = 9$ and observe that the only vertex of V^1 which is adjacent to v_{12} but lower than v_9 is v_8 – and $\rho(12, 2) = 0$. Also, it is $\rho(10, 2) = 2 - p(10) = 8$ and the only vertices of V^2 that are adjacent to v_{10} but lower than v_8 are v_2 and v_7 – and $\rho(10, 1) = 0$. Finally, it is easy to check that $\rho = 2$.

As we show in the following, it is the size of ρ that really matters for evaluating the complexity of the dynamic program of Theorem 3.2.

Theorem 3.7 *If a consistent ordering σ and a k-partition $\mathcal{V}$ of $V(G)$ are given, then a maximum weighted independent set of G may be found in $O(|V(G)| \cdot (\rho(\sigma, \mathcal{V}) + 1)^{k-1})$-time.*

Corollary 3.8 *If a consistent ordering and a k-partition of $V(G)$ are given, then a maximum weighted independent set of G may be found in $O(|V(G)| \cdot (\Delta(G) + 1)^{k-1})$-time.*

3.2 Independent sets with cardinality constraints

We here consider a slight generalization to the maximum weighted independent set problem, which will be exploited in the applications described in Section 5.1 and 5.2. As usual, we are given a graph $G(V, E)$, together with an ordering and a k-partition $V^1, \ldots, V^k$ which are consistent, but also we are given, for every class V^h, a *demand* $d^h \in \mathcal{Z}_+$. We want to find an independent set S with maximum (minimum) weight such that $| S \cap V^h |= d^h$ for $h = 1, \ldots, k$. The following theorem is a straight generalization of Theorems 3.2 and 3.7.

Theorem 3.9 *Let $G(V, E)$ be a graph together with an ordering and a k-partition $V^1, \ldots, V^k$ of V which are consistent, and $d \in \mathcal{Z}_+^k$. A maximum (minimum) weighted independent set S, such that $| S \cap V^h |= d^h$ for $h = 1, \ldots, k$, may be found in $O(|V(G)| \cdot (\rho+1)^{k-1} \cdot (D+1)^k)$-time, where $D = \max\{d^h, 1 \leq h \leq k\}$.*

4 Investigating thinness

The results in the previous section suggests the following definitions.

Definition 4.1 *A graph G is k-thin if there exists an ordering $\{v_1, v_2, \ldots, v_n\}$ and a k-partition of $V(G)$ which are consistent. The thinness of a graph G, denoted by $thin(G)$, is defined as the smallest k such that G is k-thin.*

Observe that $thin(G) \leq |V|$ for every graph $G(V, E)$. In fact,the partition of V into $|V|$ classes of size one (every class has exactly one vertex) is consistent with every ordering of V. The proof of the following proposition is straightforward.

Proposition 4.2 *The following statements hold:*

- *If H is an induced subgraph of G, then $thin(H) \leq thin(G)$.*
- *The thinness of a graph G is 1 if and only if G is an interval graph.*

4.1 Thinness and pathwidth

We here relate the thinness of a graph to its pathwidth (for a survey on the pathwidth and the treewidth, see [3]). The latter can be defined as follows:

Definition 4.3 *A path decomposition of a graph $G = (V, E)$ is a sequence of subsets of vertices $(X_1, X_2, \cdots, X_r)$ such that*

1. $\bigcup_{1 \leq i \leq r} X_i = V$
2. *for all edges $vw \in E$, there exists an i, $1 \leq i \leq r$, with $v \in X_i$ and $w \in X_i$.*
3. *for all $i, j, k \in I$: if $i \leq j \leq k$, then $X_i \cap X_k \subseteq X_j$.*

The width of a path decomposition $(X_1, X_2, \cdots, X_r)$ is defined as $\max_{1 \leq i \leq r} |X_i| - 1$. The pathwidth of a graph G is the minimum possible width over all possible path decompositions of G.

Trivially, the pathwidth of a graph G is bounded by $|V(G)| - 1$; also the bound is tight for cliques. Moreover it is easy to check that G is interval if and only if it has a path decomposition where each class is a clique. This is somehow related to the next result, giving a simple procedure for building a consistent ordering and a partition from a path decomposition.

Theorem 4.4 *Let $G(V, E)$ be a graph together with a path decomposition $(X_1, X_2, \cdots, X_r)$ of width q. There exists a consistent ordering and a k-partition of V such that $k \leq q+1$ and $\rho \leq 1$.*

There are two simple corollaries. The former is straightforward (also observe that the gap between thinness and pathwidth may be high since the thinness of a clique is 1). The latter uses Theorem 3.7 and the result was already known [3].

Corollary 4.5 *$Thinness(G) \leq Pathwidth(G) + 1$.*

Corollary 4.6 *If a path decomposition $(X_1, X_2, \cdots, X_r)$ of width q is given for a graph G, then a maximum weighted independent set of G may be found in $O(|V(G)| \cdot 2^q)$-time.*

4.2 Thinness and boxicity

Proposition 4.2 shows that k-thin graphs are a generalization of interval graphs. A different generalization is related to the definition of *boxicity* of a graph [25]. For a graph G the boxicity $b(G)$ is the minimum dimension d such that G is the intersection graph of boxes (parallelepiped) in d-dimensional space. In particular, a graph has boxicity two if and only if it is the intersection graph of rectangles. For instance, as it is shown in Figure 2, the $r \times r$ grid[2] has boxicity 2.

Graphs with boxicity one are exactly the interval graphs. In other words, a graph has boxicity 1 if and only if it has thinness 1. The next theorem shows that the thinness of a graph is lower-bounded by its boxicity.

Theorem 4.7 *$thin(G) \geq b(G)$.*

[2]For a positive integer r, the $r \times r$–grid, G_r is the graph defined by $V(G_r) = \{(i, j) : 1 \leq i, j \leq r\}$ and $E(G_r) = \{((i, j), (k, l)) : |i - k| + |j - l| = 1\}$, where $1 \leq i, j, k, l \leq r$.

One might wonder if the boxicity of a graph is equal to the thinness. Unfortunately, dealing with general graphs, the boxicity is only a lower bound for the thinness, that is, there exist graphs whose thinness is strictly larger than their boxicity. For instance, we will see in the next section that the $r \times r$ grid has thinness $\geq r/4$ (Theorem 4.10).

This is not surprising, because while a maximum weighted independent set for a k-thin graph (with fixed k) may be found in poly-time if a consistent ordering and a k-partition are given (e.g. Theorem 3.7), this is not the case dealing with boxicity. In fact, even finding a maximum independent set on the intersection graph of a (given) set rectangles is hard [12, 16].

4.3 Graphs with high thinness

In this section we show examples of graphs with high thinness and provide a general upper bound on the thinness of a graph.

It follows from Theorem 4.7 that if a graph has high boxicity it has high thinness too. An example of such a graph is the complement of a matching of size p, which has boxicity p [27] (and also the thinness of this graph is p, see Theorem 7.3 at the end of the Appendix).

We now show that the two dimensional (planar) $r \times r$ grid G_r has low boxicity (two, see Figure 2) and low degree (four), but high thinness.

Lemma 4.8 *If for a graph G there exists an integer s such that every $X \subset V$ with $|X| = s$, satisfies $|N(X)| \geq k$, then, thinness$(G) \geq \frac{k}{\Delta}$.*

The following lemma is proved in [5].

Lemma 4.9 *There exists a number s, (where $1 \leq s \leq r^2$) such that all subsets X of $V(G_r)$, with $|X| = s$, satisfy $N(X) \geq r$.*

Now, combining the above lemmas, we get that thinness$(G_r) \geq \frac{r}{4}$. Also, since the pathwidth of a $r \times r$ grid is r [24], by Corollary 4.5, thinness$(G_r) \leq r + 1$. We summarize our results in the next corollary.

Corollary 4.10 *Let G_r be the $r \times r$ grid. Then $\frac{r}{4} \leq$ thinness$(G_r) \leq r + 1$.*

We close with a couple of general upper bounds on the thinness of a graph. We point out that, while it is known that the boxicity of a graph G is bounded by $\frac{|V(G)|}{2}$ [25], we suspect this bound not hold for the thinness.

Theorem 4.11 *Let G be a graph with n vertices. Then thinness$(G) \leq n - \frac{\log n}{4}$.*

Theorem 4.12 *Let G be a graph on n vertices. Let $\Delta = \Delta(G)$ be the maximum degree of G. Then, thinness$(G) \leq n\frac{\Delta+3}{\Delta+4}$.*

5 Applications

We discuss here applications of our results in Section 3 to a pair of relevant combinatorial problem arising in wireless telecommunication systems and production system: the Frequency Assignment Problem and the Single Machine Scheduling Problem with Release Dates.

5.1 The frequency assignment problem

The Frequency Assignment Problem (FAP) is the problem of assigning radio frequencies to a set T of transmitters of a wireless network so as to establish a number of wireless connections (details on different models and approaches can be found in [1, 17]).

We are here interested in the case where, while for the rest of the network frequencies have been assigned, there is a "small" subset of transmitters $H \subset T$ where the problem has to be solved. There are two major motivation for considering this (sub)-problem: 1) the (re-)planning of small subnetworks of large operating networks is a routine operation which must be performed quite often; 2) as we show later, it is possible to design an effective solution strategy for the overall FAP, i.e. for all the transmitters T, by iteratively choosing different subsets $H \subset T$.

Namely, every transmitter $v \in H$ has to be assigned a number d_v of frequencies selected from a set $F = \{1, \ldots, f_{max}\} \subset Z_+$ of available frequencies; the transmitters in $T - H$ are supposed to be already operating and they do not require additional frequencies. In order to limit pairwise interference, the frequencies assigned to a couple (u, v) of (not necessarily distinct) transmitters of H must be "far enough" in the spectrum, i.e. they must differ by at least a suitable constant c_{uv} (usually smaller than 4). Also, assigning frequency f to transmitter $v \in H$ will cost $w_{vf} \in R_+$ to account for interference with other operating transmitters of $T - H$.

We then describe an instance of the FAP by means of a 5-tuple (H, C, d, F, w): H is a set of transmitters, $F = \{1, \ldots, f_{max}\}$ is a set of available frequencies, $C \in Z_+^{|H| \times |H|}$ is a *distance requirements* matrix, $d \in Z_+^{|H|}$ is a *demand* vector and $w \in R_+^{|H| \times |F|}$ is the frequency costs matrix.

A *feasible frequency assignment* is a family $\{F_1, \ldots, F_{|H|}\}$ of sets such that:

(i) for each $v \in H$, $F_v \subseteq F$ and $| F_v |= d_v$ (*demand constraint*);

(ii) for each $u, v \in H$, if $f \in F_u$ and $g \in F_v$, $(u, g) \neq (v, f)$, then $|g - f| \geq c_{uv}$ (*distance constraint*).

The frequency assignment problem is then the following: *find a feasible frequency assignment of H such that the sum of frequency costs is minimized.*

As we show in the following, this problem can be reduced to the solution of a minimum weight (cardinality constrained) independent set problem on a suitable graph G. In particular, it is possible to build an ordering and a partition of $V(G)$ into $|H|$ classes which are consistent. So we can use Theorem 3.9 and state the following result.

Theorem 5.1 *Let (H, C, d, F, w) be an instance of the FAP. Let $\phi = \max_{u \in H; v, z \in H} |c_{uv} - c_{uz}|$ and $D = \max_{v \in H} d_v$. Then the FAP can be solved in $O(f_{max} \cdot |H| \cdot (1+\phi)^{|H|-1} \cdot (D+1)^{|H|})$-time.*

We now shortly discuss how to approach the overall FAP problem, i.e. assign frequencies to *all* the transmitters in T [19]. In this case, *self-interference costs* has to be taken into account: these costs translate into costs over the *edges* of the conflict graph. However, small sets, named *hard sets*, of interfering transmitters with no self-interference costs can be identified, so that, dealing with hard sets, costs are only on the vertices and we have a pure independent set problem.

In the line of the ideas developed in [2, 8], hard sets can be then used to define a *large-scale neighborhood search* for the general FAP. Namely, for a hard set $H \subset T$ and an assignment s, we define the neighborhood $N_{|H|}(s)$ to be the set of all feasible assignments obtained from s by substituting *all* of the frequencies assigned to the transmitters in H and search for an optimal solution in the neighborhood. We point out that, thanks to Theorem 5.1, we can solve this optimization problem very efficiently, since usually $\phi \leq 2$.

Once we have found a better solution in the neighborhood, local search proceeds by finding optimum solutions in a sequence of neighborhoods until no improving solution exist (see [22]). The initial solutions for the local search were found by a suitable simulated annealing procedure; details on our implementation may be found in [19].

We tested our approach on the COST259 test-bed [9], a collection of large instances of the FAP in GSM networks, occasionally still operating. We were able to improve most of former best solutions, as shown in Table 6. In one case (*bradford-1-race*) the improvement exceeds 60%. Only in two cases the algorithm performed slightly worse (less than 2%). It is worth noticing that the former best solutions arise from different sources.

5.2 The Single Machine Scheduling Problem with Release Dates

The Single Machine Scheduling Problem with Release Dates (SMSR-problem) consists of scheduling a set jobs with release dates on a single machine so as to minimize the a weighted sum of the completion time of the jobs (this problem is usually referred to as $(1|r_j|\sum w_j C_j)$). The SMSR-problem is known to be NP-hard and has been widely addressed in the literature.

An instance of the SMSR-problem can be described as follows. We are given a set J of jobs to be scheduled on a single machine. For each job $j \in J$, we are given a processing time p_j, a release time r_j and a weight w_j. Let s_j be the starting time of job j in a given schedule. We want to find a *feasible* schedule, i.e. a schedule such that no more than a job is processed at each time and $s_j \leq r_j$ for any j, such that $\sum_{j \in J} w_j(s_j + p_j)$ is minimized.

As we show in the following, the SMSR-problem can be reduced to the solution of a minimum weight (cardinality constrained) independent set problem on a suitable graph G. In particular, it is possible to build an ordering and a partition of $V(G)$ which are consistent and such that $\rho = 0$ (hence the graph is interval by Corollary 3.5) so that we can use Theorem 3.9 and state:

Theorem 5.2 *If $T = \{1, 2, \dots, n\}$ define the planning horizon, then the SMSR-problem can be solved in $O(|T| \cdot 2^{|J|})$-time.*

We close by pointing out the existence of strong connections between the result above and the *time-indexed formulation* for the SMSR-problem [26]. We omit the details.

Open Questions

We close by addressing some open questions. Complexity issues are open. We recall that recognizing the boxicity of a graph is NP-hard: this was shown in three steps. First, Cozzens [6] showed that computing the boxicity of a graph is NP-hard; this was improved by Yannakakis [28] to testing whether $b(G) \leq 3$ is NP-complete and by Kratochvil [18] to determining whether $b(G) \leq 2$ is NP-complete. We suspect that similar results hold for k-thin graphs, but a proper proof has eluded our attempts so far.

A number of interesting issues concerns the links between thinness and boxicity. Theorem 4.7 shows that the thinness of a graph is less or equal than its boxicity. It would be interesting to characterize graphs for which the boxicity is equal to the thinness, and, in particular, cases, where, from a box representation of a graph, it is possible to build a consistent ordering and partitions as to use the results of Section 3.

References

[1] K. Aardal, S.P.M. van Hoesel, A. Koster, C. Mannino, and A. Sassano, Models and Solution Techniques for Frequency Assignment Problems, 4OR , (1) 4, 261-317, 2003.

[2] R. K. Ahuja, O. Ergun, J. B. Orlin and A. P. Punnen, *A survey of Very Large-Scale Neighborhood search Techniques*, http://web.mit.edu/jorlin/www/papersfolder/VLSN.pdf.

[3] H. Bodlaender, *A Touriste Guide Through Treewidth*, Acta Cybernetica, 11, 1993, 1-21.

[4] K. Booth and S. Leuker, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J.Comput. System. Sci., 13, 1976, 335-379.

[5] J. Chvátalová, *Optimal Labelling of a product of two paths* Discrete Mathematics, 11.3, 1975 249–253.

[6] M.B. Cozzens, *Higher and multidimensional analogues of interval graphs*, Ph.D. Thesis, Rutgers University, New Brunswick, NJ, 1981.

[7] M.B. Cozzens, F.S. Roberts, *Computing the boxicity of a graph by covering its complement by cointerval graphs*, Discrete Applied Mathematics 6, 1983, 217-228.

[8] V.G. Deineko, G.J. Woeginger, *A study of exponential neighborhoods for the Travelling Salesman Problem and for the Quadratic Assignment Problem*, Mathematical Programming 87 (A), 2000, 519-542.

[9] A. Eisenblätter and A. Koster, *FAP web - A website about Frequency Assignment Problems*, http://fap.zib.de/.

[10] P.C. Fishburn, *Interval Orders and Interval Graphs*, John Wiley, New York, 1985.

[11] M. Formann, F. Wagner, *A packing problem with application to lettering of maps*, in Proceedings of the 7-th Annual ACM Symposium in Compuational Geometry, 1991, 281-288.

[12] R.J. Fowler, M.S. Paterson and S.L. Tanimoto, *Optimal packing and covering in the plane are* NP*-complete*, Information Processing Letters 12(3), 1981, 133-137.

[13] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[14] U.I. Gupta, D.T. Lee and J.Y.T Leung *Efficient Algorithms for Interval Graphs and Circular-Arc Graphs*, Networks, 12, 1982, 459-467.

[15] G. Hajos, *Uber eine Art von Graphen*, International Math. Nachr. 11, 1957, Problem 65.

[16] H. Imai and T. Asano, *Finding the connected component and a maximum clique of an intersection graph of rectangles in the plane*, Journal of Algorithms 4, 1983, 310-323.

[17] A. Koster. *Frequency Assignment - Models and Algorithms*, Ph.D. Thesis, Maastricht University, Netherlands 1999.

[18] J. Kratochvil, *A Special Planar Satisfiability Problem and a consequences of its* NP*-completeness*, Discrete Applied Mathematics 52, 1994, 233-252.

[19] C. Mannino, G. Oriolo and F. Ricci, *Solving Large Instances of Frequency Assignment*, Tech. Report 22-02 Dipartimento di Informatica e Sistemistica, Universitá di Roma *La Sapienza*, 2002.

[20] S. Olariu *An optimal greedy heuristic to color interval graphs*, Information Processing Letters 37, 1991, 21-25.

[21] R.J. Opsut, F.S. Roberts, *On the fleet maintenance, mobile radio frequency, task assignment, and traffic phasing problems*, G. Chartand et al. eds., The Theory and Application of Graphs, Wiley, 1981, 479-492.

[22] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, 1982.

[23] J. Ramalingam and C. Pandu Rangan, *A unified approach to domination problems on interval graphs*, Information Processing Letters 27, 1988, 271-274.

[24] N. Robertson, P.D. Seymour *Graph Minors. X. Obstructions to tree Decompositions*, Journal Combinatorial Theory Series B 52, 1991, 152-190.

[25] F.S. Roberts, *On the boxicity and cubicity of a graph*, in: W.T.Tutte, ed., Recent Progress in Combinatorics, Academic Press, New York, 1969, 301-310.

[26] J. Sousa, L.A. Wolsey *Time-indexed formulations of non-preemptive single machine scheduling problems*, Mathematical Programming 54, 1992, 353-367.

[27] W.T. Trotter, Jr., *A forbidden subgraph characterization of Robert's inequality for boxicity*, Discrete Mathematics 28, 1979, 303-314.

[28] M. Yannakakis, *The complexity of the partial order dimension problem*, SIAM Journal on Algebraic Discrete Methods, 3, 1982, 351-358.

6 Appendix A: FIGURES AND TABLES

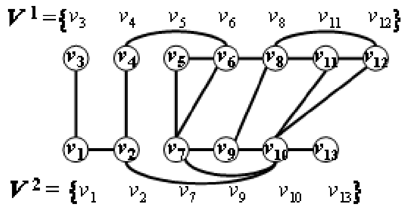

Figure 1: *A graph along with an ordering and a 2-partition of the vertices which are consistent. The ordering is* $V = \{v_1, v_2, \ldots, v_{13}\}$ *and the classes of 2-partition are* $V^1 = \{v_3, v_4, v_5, v_6, v_8, v_{11}, v_{12}\}$ *and* $V^2 = \{v_1, v_2, v_7, v_9, v_{10}, v_{13}\}$

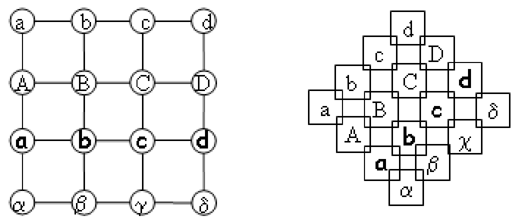

Figure 2: *The A* 4×4 *grid has boxicity 2 but high thinness (Corollary 4.10)*

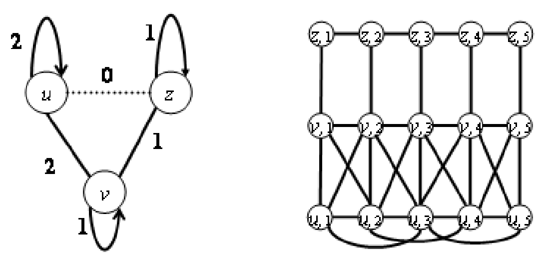

Figure 3: *An instance of FAP with* $c_{uz} = 0$, $c_{vv} = c_{zz} = c_{vz} = 1$, $c_{uv} = c_{uu} = 2$ *and the associated conflict graph*

Instance name	Size	Old Best	SA + N_Z	Time
siemens1	930	2.30	2.20	5
siemens2	977	14.28	14.27	9
siemens3	1623	5.19	5.1	15
siemens4	2785	80.97	77.24	18
bradford_nt-1-eplus	1970	0.86	0.86	22
bradford_nt-1-free	1970	0	0	17
bradford_nt-1-race	1970	0	0	16
bradford_nt-2-eplus	2199	3.17	3.20	24
bradford_nt-2-free	2199	0	0	12
bradford_nt-2-race	2199	0	0	13
bradford_nt-4-eplus	2650	17.73	17.72	21
bradford_nt-4-free	2650	0	0	12
bradford_nt-4-race	2650	0	0	11
bradford_nt-10-eplus	2468	146.20	144.94	19
bradford_nt-10-free	2468	5.86	5.42	14
bradford_nt-10-race	2468	1.07	1.09	14
bradford-0-eplus	2947	0.80	0.60	64
bradford-0-free	2947	0	0	30
bradford-0-race	2947	0	0	26
bradford-1-eplus	2947	33.99	33.80	64
bradford-1-free	2947	0.16	0.12	30
bradford-1-race	2947	0.03	0.01	26
bradford-2-eplus	3406	80.03	79.38	75
bradford-2-free	3406	2.95	2.69	37
bradford-2-race	3406	0.42	0.32	39
bradford-4-eplus	3996	167.7	167.0	90
bradford-4-free	3996	22.09	20.00	46
bradford-4-race	3996	3.04	2.93	36
bradford-10-eplus	4871	400.0	395.5	128
bradford-10-free	4871	117.8	113.7	63
bradford-10-race	4871	30.22	27.38	46

Table 1: Computational results over COST253 instances. Columns: *Size* number of assigned frequencies; *Old Best* value of best known solution, $SA + N_Z$ value of our solution, $Time$ time to best in hours. All solutions and references are available at the FAP-WEB [9]. The times refer to a C code, running under LINUX operating system on a two processors Intel Xeon, 1.5 GHz.

7 Appendix B: PROOFS

Theorem 3.2

Proof. Let $G = (V, E)$ be a graph with weights $w(v)$ for each vertex $v \in V$. Let $\{v_1, \ldots v_n\}$ and $(V^1, \ldots, V^k)$ be an ordering and a partition of V which are consistent. Finally assume that $V^h = \{v_1^h, \ldots, v_{p^h}^h\}$, with $v_1^h < v_2^h < \ldots < v_{p^h}^h$.

Each state of the dynamic program is associated to a k-tuple $(j^1, \ldots, j^k)$ with $0 \leq j^h \leq p^h$ for each h; the set of all such k-tuples is denoted by $\mathcal{K}$. Let $\alpha_w(0, \ldots, 0) = 0$ and, for each $(j^1, \ldots, j^k) \in \mathcal{K}$ denote by $\alpha_w(j^1, \ldots, j^k)$ the maximum weight of an independent set $S_{(j^1,\ldots,j^k)} \subseteq \bigcup_{h=1..k}\{v_1^h, \ldots, v_{j^h}^h\}$.

We want to evaluate $\alpha_w(j^1, \ldots, j^k)$. Without loss of generality, we assume that $u = v_{j^1}^1 > v_{j^h}^h$ for $h \in \{2, \ldots, k\}$. As usual, either $u \notin S_{(j^1,\ldots,j^k)}$ or $u \in S_{(j^1,\ldots,j^k)}$. In the former case, $S_{(j^1,\ldots,j^k)}$ is also a maximum weighted independent set in $G[\{v_1^1, \ldots, v_{j^1-1}^1\} \cup_{h=2..k} \{v_1^h, \ldots, v_{j^h}^h\}]$. In the latter case, $S_{(j^1,\ldots,j^k)} - u$ is also a maximum weighted independent set in $G[\bigcup_{h=1..k}\{v_1^h, \ldots, v_{j^h}^h\} - N(u)]$.

Claim. *For $1 \leq h \leq p$, $\{v_1^h, \ldots, v_{j^h}^h\} - N(u) = \{v_1^h, \ldots, v_{b^h}^h\}$, with $b^h = \min(j^h, |\overline{N}(u,h)_<|)$.*

By Lemma 3.1, since the ordering and the partition are consistent, for any class V^h we have that $\overline{N}(u,h)_< = \{v_1^h, v_2^h, \ldots, v_{|\overline{N}(u,h)_<|}^h\}$. Also, since u is the highest vertex in the set $\{v_{j^1}^1, \ldots, v_{j^k}^k\}$, for each h we have that $\{v_1^h, \ldots, v_{j^h}^h\} - N(u) = \{v_1^h, \ldots, v_{j^h}^h\} \cap \overline{N}(u,h)_< = \{v_1^h, v_2^h, \ldots, v_{b^h}^h\}$, where $b^h = \min(j^h, |\overline{N}(u,h)_<|)$. ■

Therefore, the following recursion holds for $\alpha_w(j^1, \ldots, j^k)$:

$$\alpha_w(j^1, ..., j^k) = \max \begin{cases} \alpha_w(j^1 - 1, j^2, j^3, \ldots j^k) \\ w(u) + \alpha_w(b^1, \ldots, b^k) \end{cases} \tag{4}$$

Since $b^h \leq j^h$ for every h and $b^1 < j^1$, both k-tuples $(j^1 - 1, j^2, \ldots, j^k)$ and $(b^1, \ldots, b^k)$ are dominated by $(j^1, \ldots, j^k)$ and are in $\mathcal{K}$. The recursion will generate in a finite number of steps the (initial) k-tuple $(0, \ldots, 0)$. This fact, along with the initial condition $\alpha_w(0, \ldots, 0) = 0$ ensures that the recursion terminates.

Finally, since $\alpha_w(G) = \alpha_w(p^1, \ldots, p^k)$ and the size $|\mathcal{K}|$ of the set of k-tuples is equal to $(p^1 + 1)(p^2 + 1) \ldots (p^k + 1) \leq (\frac{|V|}{k} + 1)^k$, a maximum independent set on G may be found in $O(\frac{|V|}{k})^k$-time. □

Theorem 3.7

Proof. Recall that we define a k-tuple $(j^1, \ldots, j^k)$ to be feasible if $\alpha_w(j^1, \ldots, j^k)$ has to be evaluated in order to compute $\alpha_w(G) = \alpha_w(p^1, \ldots, p^k)$. In order to prove the statement it is therefore enough to prove the following:

Claim. *Let $v \in V(G)$. The number of feasible k-tuple $(j^1, \ldots, j^k)$ such that v is the highest vertex in the set $\{v_{j^1}^1, \ldots, v_{j^k}^k\}$ is bounded by $(\rho + 1)^{k-1}$.*

For sake of simplicity, we prove the claim for the case $k = 2$: the extension to the case $k > 2$ is direct. Hence, we are given an ordering $\{v_1, \ldots, v_n\}$ and a partition (V^1, V^2) of V which are consistent. We assume that $V^1 = \{u_1, \ldots, u_l\}$ and $V^2 = \{z_1, \ldots, z_m\}$. Also we are given weights $w(v)$ for each vertex $v \in V$. As in the proof of Proposition 3.4, for each vertex $v = v_f$

and class V^h, we denote by $q(v,h)$ be the number of vertices in each class not larger than v, i.e. $q(v,h) = |V^h \cap \{v_1, \ldots, v_f\}|$.

So suppose that (a,b) is a feasible pair and $v \in V(G)$ is the highest vertex in the set $\{u_a, z_b\}$. We will show that a and b cannot be "far" from $q(v,1)$ and $q(v,2)$. Observe that by definition $q(v,1) \geq a$ and $q(v,2) \geq b$. In particular, if $v = u_a$ then $q(v,1) = a$, while if $v = z_b$ then $q(v,2) = b$; moreover, we will show that the following holds:

$$\text{if } v = u_a \text{ then } 0 \leq q(v,2) - b \leq \rho; \text{ if } v = z_b \text{ then } 0 \leq q(v,1) - a \leq \rho \qquad (*)$$

which, of course, implies our claim.

The proof is by induction. Observe that $(*)$ holds for $(a,b) = (l,m)$. In fact, both $v = u_l$ and $v = z_m$ imply $q(v,1) = l$, $q(v,2) = m$ ... (0,0).

We now assume that $(*)$ holds for the current pair (a,b) and show that the same holds for the terms arising from (4). We assume without loss of generality that $u_a > z_b$; also we let $r = \overline{N}(u_a,1)_<$ and $s = \overline{N}(u_a,2)_<$. Finally we rewrite (4) as follows:

$$\alpha_w(a,b) = \max \begin{cases} \alpha_w(a-1,b) \\ w(u_a) + \begin{cases} \alpha_w(r,s) & \text{if } s < b \\ \alpha_w(r,b) & \text{if } s \geq b \end{cases} \end{cases} \tag{5}$$

Recall that by assumption we have that $q(u_a,2) - b \leq \rho(u_a,2)$.

Consider the pair $(a-1,b)$ coming from the first term of the recursive equation (5). If $u_{a-1} > z_b$, then $q(u_{a-1},2) = b \leq q(u_a,2) - b \leq \rho(u_a,2)$. If $u_{a-1} < z_b < u_a$, then $q(z_b,1) = a-1$.

Now consider the other term of the recursion. First suppose that $s < b$ and consider therefore the pair (r,s). If $u_r > z_s$, then $q(u_r,2) - s = \rho(u_a,2)$ by definition. If $u_r < z_s$ then $q(z_s,1) - r = \rho(u_a,1)$ by definition.

Finally, suppose that $s \geq b$ and consider therefore the pair (r,b). If $u_r > z_b$, then $q(u_r,2) - b \leq q(u_a,2) - b \leq \rho(u_a,2)$. If $u_r < z_b$, then $q(z_b,1) - r \leq q(z_s,1) - r = \rho(u_a,1)$ by definition. ■

□

Theorem 3.9

Proof. (Sketch). Equation (4) can be generalized to solve this problem. Assume that, for every $1 \leq h \leq k$, $V^h = \{v_1^h, \ldots, v_{p^h}^h\}$; denote by $\mathcal{K}$ the set of the k-tuples $(j^1, \ldots, j^k)$ such that $0 \leq j^h \leq p^h$ and by $\mathcal{D}$ the set of the k-tuples $(r^1, \ldots, r^k)$ such that $0 \leq r^h \leq d^h$.

For each $(j^1, \ldots, j^k) \in \mathcal{K}$ and $(r^1, \ldots, r^k) \in \mathcal{D}$, denote by $\alpha_w(j^1, \ldots, j^k; r^1, \ldots, r^k)$ the maximum (minimum) weight of an independent set S such that $| S \cap V^h |= r^h$ and $S \cap V^h \subseteq \{v_1^h, \ldots, v_{j^h}^h\}$.

If $(j^1, \ldots, j^k) \neq (0, \ldots, 0) \in \mathcal{K}$, $(r^1, \ldots, r^k) \neq (0, \ldots, 0) \in \mathcal{R}$ and, without loss of generality, $u = v_{j^1}^1 > v_{j^h}^h$ for $h \in \{2, \ldots, k\}$ and $r^1 > 0$, then the following dynamic programming equation holds (replace max with min for an independent set with minimum weight):

$$\alpha_w(j^1, \ldots, j^k; r^1, \ldots, r^k) = \max \begin{cases} \alpha_w(j^1 - 1, j^2, \ldots, j^k; r^1, \ldots, r^k) \\ \alpha_w(b^1, \ldots, b^k; r^1 - 1, r^2, \ldots, r^k) + w(u) \end{cases} \tag{6}$$

where the vector b is defined as for equation (4). Then we proceed as with Theorem 3.7. We omit the details. □

Theorem 4.4

Proof. Let $X_1, X_2, \cdots, X_r$ be the subsets in the order appearing in the path decomposition. We first describe an ordering and then give a description of how we can assign the vertices to classes, in order to get a k-partition consistent with the ordering.

(1) **Ordering.** Let x be a vertex. Let i be the smallest integer such that $x \in X_i$. Then we say that x is first-encountered in X_i. Our numbering is as follows. If x is first encountered in X_i and y is first encountered in X_j with $i < j$ then $x < y$. If $i = j$ then x and y can be relatively put in arbitrary order.

(2) **k-partition.** We start with X_1 and process the vertices in it. The rule is, we assign each vertex of X_1 to a different class. Now we take X_2 and we do the same thing. Note that some vertices in X_2 are already assigned to certain classes, but all different classes. We have only to deal with the remaining vertices of X_2, namely the vertices of $X_2 - X_1$: we assign them to classes not used by the vertices of $X_2 \cap X_1$. We carry on with this procedure: at each step i we have only to assign vertices of $X_i - X_{i-1}$ to classes not used by the vertices of $X_i \cap X_{i-1}$. Clearly since $|X_i| \leq q$ for any i, we can implement this procedure so that the number of k of classes we built is less or equal than q.

Also we claim that by the process, for each i, vertices in X_i are assigned to classes such that if $x, y \in X_i$, then x and y goes to different classes. Clearly, it is enough to show that when we reach X_i, those vertices in X_i which are already assigned are in different classes. If not, let us assume that i is the smallest integer such that inconsistency occurs in X_i. Let x and y be the two vertices in X_i, which happen to be in the same class at this stage. Clearly both x and y are first-encountered in X_j and X_k , with $j < k < i$ (without loss of generality). Then it means x, y have together occurred before in some subset earlier than X_i, for example in X_k (y is in X_k by assumption; x is in X_k because it is in X_j as well as in X_i: then, by the definition of pathwidth $x \in X_k$ also.) This contradicts the fact that inconsistency was occurring for the first time in X_i.

So finally, we write down what we have achieved: If $x, y \in X_i$, for some i, then class(x) $\neq$ class(y).

Claim. *Let the vertex x be first-encountered in X_i. Then, all neighbors of x lower than x are also present in X_i. (Just present - not necessarily "first encountered").*

Let y be such a vertex. If it is not in X_i, then it was first found in some X_j where $j < i$, since $y < x$. But since xy is an edge there should be one subset X_k such that such the both $x, y \in X_k$. But since i is the smallest number such that $x \in X_i$, $k > i$. That is $y \in X_j$ and $y \in X_k$ where $j < i$ and $k > i$ and $y \notin X_i$. This contradicts with the definition of pathwidth. ■

So, all neighbors of x which are lower than x are present in X_i. Thus neighbors of x, which are smaller than x, go into different classes. Now consider the class in which a neighbor $y < x$ is present. We just need to ensure that there exists no other vertex z in the same class such that $y < z < x$. Suppose there exists one such vertex z. Note that if z and y are in the same class, there is no subset X_k such that z, y both in X_k. Thus $y < z$ tells us that y is first encountered in X_k and z is first encountered in X_j with $k < j$. Also, $z < x$ tells us that $j \leq i$. But by the Claim above $y \in X_i$, since y is a neighbor of x (and is lower than x). Thus $y \in X_k$, $y \in X_i$, but not in X_j where $k < j \leq i$, which is impossible by the definition of pathwidth.

Thus, the k-partition and the ordering of $V(G)$ we obtained are consistent. Finally observe that, since neighbors of x, which are smaller than x, go into different classes, then in each class there is at most one neighbor of x which is smaller than x. It follows that $\rho \leq 1$. □

Theorem 4.7

Proof. It has been observed in [7] that a graph $G(V,E)$ has boxicity k if and only if $G(V,E)$ is the *intersection* of k interval graphs $G^1(V,E^1)$, $G^2(V,E^2)$, ... , $G^k(V,E^k)$ with the same vertex set as G - i.e. $G = G(V, E^1 \cap E^2 \cap ... \cap E^k)$.

Let $G(V,E)$ be a graph with thinness k. By definition, there exists an ordering $<$ and a k-partition $V^1, \dots, V^k$ of V which are consistent. For each $1 \leq h \leq k$, let $V^h = \{v^h_1, \dots, v^h_{p^h}\}$ and define an interval graph $I^h(V,E^h)$ as follows (remark that I^h has the same vertex set of G):

(i) $I^h[V^h] = G[V^h]$, i.e. for all $u,v \in V^h$, $uv \in E^h$ iff $uv \in E$.

(ii) $I^h[V \setminus V^h]$ is a complete graph, i.e. for all $u,v \notin V^h$, $uv \in E^h$.

(iii) For each $u \in V \setminus V^h$ and let r be the smallest index such $uv^h_r \in E$. Then $uv^h_i \in E^h$ for $i = r, \dots, p^h$.

We claim that $G = I^1 \cap \cdots \cap I^k$ and that every I^h is an interval graph. It follows that $b(G) \leq k$.

We first show that I^h is an interval graph. To this end, let $\{w_1, \dots, w_n\}$ be an ordering of the vertices of I^h such that $w_1 = v^h_1, \dots, w_{p^h} = v^h_{p^h}$, i.e. the first p^h vertices coincide with the vertices in V^h while the remaining vertices are randomly ordered. We show that if $r < s < t$ and $w_t w_r \in E^h$ then $w_t w_s \in E^h$. In fact, if $w_t, w_s \notin V^h$ this holds by (ii). If $w_t \in V^h$ then also $w_r, w_s \in V^h$ and the result follows by (i) and by the fact that $G[V^h]$ is 1-thin. Finally if $w_t \notin V^h$ and $w_s \in V^h$ then $v_r \in V^h$ and the result follows by (iii). So, by Theorem 7.3, I^h is interval.

We show now that $G = I^1 \cap \cdots \cap I^k$. In particular, we need to show that (j) if $uw \in E$ then $uw \in E^h$ for $h = 1, \dots, k$ and (jj) if $uw \in E^h$ for $h = 1, \dots, k$ then $uw \in E$.

(j). Suppose $uw \in E$. If $u, w \in V^h$ for some $h \in \{1, \dots, k\}$ then $uw \in E^h$ by (i) and $uw \in E^q$ for $q \neq h$ by (ii). If $u \in V^h$ and $w \in V^t$ with $t \neq h$, then $uw \in E^h$ and $uw \in E^t$ by (iii), while $uw \in E^q$ for $q \neq h, t$ by (ii).

(jj). Suppose now $uw \in E^h$ for $h = 1, \dots, k$. Suppose w.l.o.g. that $w < u$ in the global ordering and let $w \in V^j$. Recall that $V^j = \{v^j_1, \dots, v^j_{p^j}\}$ and let $w = v^j_m$. Since $uw \in E^j$, by (iii) there exists an index $l \leq m$ such that vertex $uv^j_l \in E$. If $l = m$ then $uv^j_l = uv^j_m = uw \in E$, while if $l < m$ then $v^j_l < v^j_m = w < u$ and consistency implies $uw \in E$. □

Lemma 4.8

Proof. Let the thinness of G be t. Then there exists a partition $V^1, \dots, V^t$ and an ordering $\{v_1, \dots v_n\}$ of $V(G)$ that are consistent. Let $S = \{v_{n-s+1}, \dots v_n\}$, that is S is the set of the s highest vertices of G. Note that, since every node in $N(S)$ is outside S, each of them is lower than any vertex in S.

We claim that for each i, $1 \leq i \leq t$, $|V^i \cap N(S)| \leq \Delta$. (That is none of the classes can contain more than Δ vertices from $N(S)$.) Suppose V^i contains at least $\Delta + 1$ vertices from $N(S)$. Let x be the lowest vertex in $V^i \cap N(S)$. Clearly x is adjacent to some vertex $y \in S$, since $x \in N(S)$. Then, y has to be adjacent to all the vertices in $V^i \cap N(S)$, since all of them are lower than y. Thus, the degree of y has to be at least $\Delta + 1$, contradicting the fact that the maximum degree of G is Δ.

So, each class contains only at most Δ vertices of $N(S)$. It follows that there are at least $\frac{|N(S)|}{\Delta} \geq \frac{k}{\Delta}$ classes. □

Theorem 4.11
Proof. We first need a definition and a lemma whose proof is straightforward.

Definition 7.1 *Let $G(V,E)$ be a graph. A partition $V^1,\ldots,V^k$ of $V(G)$ is valid if there exists an ordering which is consistent with this partition. Moreover, a class V^i is called a singleton class if $|V^i|=1$; otherwise V^i is a non–singleton class.*

Lemma 7.2 *Let $G(V,E)$ be a graph with a valid partition $V^1,\ldots,V^k$ of $V(G)$. Let $X=\{v\in V : v$ belongs to a singleton class$\}$. Then there exists an ordering $\{v_1,\ldots v_n\}$ of $V(G)$ which is consistent with the partition and is such that $X=\{v_1,\ldots v_{|X|}\}$, i.e., vertices of X are the lowest.*

Let t be the thinness of G. Then there exists a valid partition of $V(G)$ using t classes. We can assume that there are at least 2 singleton classes in this valid partition. Otherwise, if every class (except possibly one), contains at least 2 vertices, then clearly $t\le\frac{n}{2}+1\le n-\frac{\log n}{4}$, as required by the Theorem.

Let $X=\{u\in V : u$ belongs to a singleton class$\}$. Let $|X|=h$ and $|V-X|=q$ (clearly, $h+q=n$).

Claim. $h\le 2^q$.

Let $V-X=\{y_1,y_2,\cdots,y_q\}$. We define a function $f: X\to\{0,1\}^q$ as follows:
For $x\in X$, $f(x)=(a_1,a_2,\cdots,a_q)$, where $a_i=1$, if x is adjacent to y_i, and $a_i=0$ otherwise.

Assume for contradiction that $h>2^q$. Then clearly there exists two vertices $r,s\in X$, such that $f(r)=f(s)$ (since $|\{0,1\}^q|=2^q$). Now, we claim that even if we merge the two classes containing r and s into one class, it will remain a valid partition of $V(G)$. This will provide the required contradiction, since we have assumed that t is the thinness of G.

By Lemma 7.2, there exists an ordering $\{v_1,\ldots v_n\}$ which is consistent with the given partition, such that the vertices of the singleton classes are the lowest, i.e. $X=\{v_1,\ldots v_{|X|}\}$. In fact, it is possible to further assume (without violating the validity of the partition) that the $r=v_{h-1}$ and $s=v_h$.

Now, consider merging the two classes containing r and s into one class. It is not difficult to verify that the resulting partition is still valid since the previous ordering is still consistent: Clearly there is no conflict due to vertices of $X-\{r,s\}$, since they are all numbered lower than $r=v_{h-1}$. Also, since $f(r)=f(s)$, it is easy to see (from the definition of the function f) that for any vertex $y\in V-X$, y is adjacent to r if and only if it is adjacent to s. Therefore there can not be any conflict due to the vertices of $V-X$. Thus we have a valid partition of $V(G)$ using only $t-1$ classes, contradicting the assumption that the thinness of G is t. We infer that $h\le 2^q$. ■

Now, suppose $h>n-\frac{\log n}{2}$. Then from $2^q\ge h$, we get $q\ge\log(n-\frac{\log n}{2})$. But this leads to a contradiction since $|X|+|V-X|=h+q\ge n-\frac{\log n}{2}+\log(n-\frac{\log n}{2})>n$. So, we infer that $h\le n-\frac{\log n}{2}$. Then, $q=n-h\ge\frac{\log n}{2}$. Note that $t\le h+\frac{q}{2}$, since every non–singleton class contains at least 2 vertices, and there are only a total of q vertices in non–singleton classes. Thus, $t\le h+q-\frac{q}{2}=n-\frac{q}{2}\le n-\frac{\log n}{4}$, as required by the Theorem. □

Theorem 4.12
Proof. We modify the proof of Theorem 4.11 as follows.

Claim. $h \leq \frac{\Delta+2}{2}k$.

For $x \in X$, we denote by $|f(x)|$ the number of 1 s in the q–tuple $f(x)$. Clearly, $\sum_{x \in X} |f(x)| \leq \sum_{y \in V-X} degree(y) \leq q\Delta$. Now partition X into two classes as follows: Let $X_1 = \{x \in X : |f(x)| = 1\}$ and $X_2 = X - X_2$.

Note that $|X_1| \leq q$ since otherwise if $|X_1| > q$, there will be two vertices $r, s \in X_1$ such that $f(r) = f(s)$, leading to contradiction, as described in the proof of Theorem 4.11. We can assume that there is no vertex $x \in X_2$ with $|f(x)| = 0$, since if such a vertex existed we could have merged the class of this vertex with that of the highest vertex in X, thus getting a contradiction. Thus for every vertex $x \in X_2$, $|f(x)| \geq 2$. Thus, since $\sum_{x \in X_2} |f(x)| \leq \sum_{x \in X} |f(x)| \leq q\Delta$, we have $|X_2| \leq \frac{q\Delta}{2}$. Thus, $h = |X| = |X_1| + |X_2| \leq q\frac{\Delta+2}{2}$. ■

It follows that $q = n - h \geq n - q\frac{\Delta+2}{2}$. Rearranging, we get $q \geq n\frac{2}{\Delta+4}$. Now, $t \leq h + \frac{q}{2} = n - \frac{q}{2} \leq n\frac{\Delta+3}{\Delta+4}$. □

Theorem 5.1

Proof. The graph $G(W, E)$ is defined as follows:

$W = \{(v, f) : v \in H, f \in F\}$: there exists a vertex of G for every pair (v, f) where v is a transmitter and f is a frequency;

$E = \{(u, g)(v, f) : |g - f| < c_{uv}, (u, g) \text{ and } (v, f) \in W, (u, g) \neq (v, f)\}$: an edge $(u, g)(v, f) \in E$ indicates that we cannot (simultaneously) assign frequency g to u and frequency f to v.

Graph G is called the *conflict graph* [1]. With every subset of vertices $S \subseteq W$ we immediately associate a frequency assignment $\mathcal{F} = \{F_1, \ldots, F_{|H|}\}$, by letting $F_v = \{f : (v, f) \in S\}$ for all $v \in H$. Moreover, if S is an independent set of G, then S corresponds to feasible assignment of frequencies to the transmitters of H whose cost is equal to the weight $w(S) = \sum_{(v,f) \in S} w_{v,f}$: similarly, every feasible assignment corresponds to an independent set of G of equal weight. So, denoting by $W^v = \{(v, f) : f \in F\}$, finding a minimum cost feasible frequency assignment of H is equivalent to finding an independent set S of $G(W, E)$ such that $|S \cap W^v| = d_v$ for each $v \in H$ and the weight of S is minimum.

Claim. *The conflict graph $G(W, E)$ is $|H|$-thin.*

Let $H = \{v_1, \ldots, v_n\}$. Define on the set W the ordering $\sigma = \{(v_1, 1), \ldots, (v_n, 1), \ldots, (v_1, f_{max}), \ldots, (v_n, f_{max})\}$ and the partition $\mathcal{W} = \{W^{v_1}, \ldots, W^{v_n}\}$. It is immediate to verify that σ and $\mathcal{W}$ are consistent. ■

So, our original task is now reduced to the problem of finding a minimum weight (cardinality constrained) independent set problem on a $|H|$-thin graph. To complete the proof we only need to verify that $\rho(\sigma, \mathcal{W}) = \phi$ and the result follows immediately from Theorem 3.9. In order to evaluate $\rho(\sigma, \mathcal{W})$, we need to compute, for every $v_j \in W$, the quantity $\rho(v_j, z)$, i.e. the number of vertices of W^z that are smaller than $v_{p(j)}$ and are adjacent to v_j. First observe that the ordering σ implies $(x, r) < (u, f)$ whenever $r < f$. Now, let $v_j = (u, f)$. If $\overline{N}(v_j, z)_<$ is empty then $\rho(v_j, z) = 0$; otherwise $v_{p(j)} = max_{h<f}\{(w, h) : f - h = c_{uw}\}$. Suppose the maximum is attained for $v_{p(j)} = (v, f - c_{uv})$. Observe now that it is $f - g < c_{uz}$ for every vertex (z, g) adjacent to (u, f). In particular, for a class W^z, the vertices of W^z which are adjacent to (u, f) belong to the set $\{(z, f - c_{uz} + 1), \ldots, (z, f)\}$. Also, the vertices in the set $\{(z, f - c_{uv} + 1), \ldots, (z, f)\}$ are higher then $v_{p(j)} = (v, f - c_{uv})$. So, the vertices in W^z that are adjacent to $v_j = (u, f)$ but smaller than $v_{p(j)} = (v, f - c_{uv})$ belong to the set $\{(z, f - c_{uz} + 1), \ldots, (z, f - c_{uv})\}$ and thus their cardinality $\rho(v_j, z)$ is at most $c_{uz} - c_{uv}$. The result follows from the definition of ρ.

□

Theorem 5.2
Proof. The graph $G(W, E)$ is defined as follows:

$W = \{(j,t) : j \in J, t \in T\}$: there exists a vertex of G for every pair (j,t) where j is a job and t an instant of the time horizon. Every vertex (j,t) can thus be interpreted as a potential *starting time* of job j.

$E = \{(u,g)(v,f) : (u,g) \neq (v,f)$ and either (i) $g \geq f$ and $g - f < p_v$; or (ii) $f \geq g$ and $f - g < p_u\}$. That is, an edge $(u,g)(v,f) \in E$ indicates that we cannot start job v (job u) at time f (time g) if we started a job u (job v) at time g (time f).

Observe that every feasible assignment of jobs to starting times corresponds to an independent set S such that $|S \cap W_j| = 1$ for $j \in J$, where $W_j = \{(j,t) : t \in T\}$, for each $j \in J$. In other words, if we give to every vertex (j,t) the weight:

$$w(j,t) = \begin{cases} M & \text{if } t < R_j \text{ or } t + p_j > |T| \\ w_j(t + p_j) & \text{otherwise} \end{cases} \tag{7}$$

with M large positive constant, then the SMSR-problem can be solved by finding an independent set S with minimum weight such that $|S \cap W_j| = 1$ for $j \in J$.

Claim. *There exist an ordering and a partition of W which are consistent and such that $\rho = 0$.*

Let $J = \{1, \ldots, |J|\}$. Define on the set W the ordering $\sigma = \{(1,1), \ldots, (|J|,1), \ldots, (1,T), \ldots, (|J|,T)\}$ and the partition $\mathcal{W} = \{W_1, \ldots, W_{|J|}\}$. It is immediate to verify that σ and $\mathcal{W}$ are consistent. Finally it is easy to check that, $\rho(\sigma, \mathcal{W}) = 0$. ■

Hence, by applying Theorem 3.9, the SMSR-problem can be solved in $O(|T|2^{|J|})$-time. □

Theorem 7.3 *Let T^n be the graph with n vertices and $(n-2)$-regular (the complement of T^n is a perfect matching). The thinness of T^n is $n/2$.*

Proof. Let the vertex set of T^n be equal to $\{x_1, y_1, \ldots, x_{n/2}, y_{n/2}\}$ and suppose that (x_i, y_i), for $1 \leq i \leq n/2$, are the only pairs of non-adjacent vertices.

If we define, for $1 \leq i \leq n/2$, $V^i = \{x_i, y_i\}$, then every total order on the vertices of $V(T^n)$ is consistent with this partition. We now show that T^n is not $(n/2 - 1)$-thin.

Suppose the contrary, that is there exists an ordering $>$ on the vertices of $V(T^n)$ and a partition of $V(T^n)$ in $(n/2 - 1)$ classes $(V^1, \ldots, V^{(n/2-1)})$ which are consistent.

For every class, denote by $f(V^h)$ the smallest element of V^h with respect to the ordering $>$. Clearly, there exists at least one pair $\{x_i, y_i\}$, $1 \leq i \leq n/2$, such that $\bigcup_h f(V^h) \cap \{x_i, y_i\} = \emptyset$. Without loss of generality, assume that such pair is (x_1, y_1) and that $y_1 > x_1$.

Let V^q be the class which x_1 belongs to. It follows that y_1 is adjacent to $f(V^q)$ and non-adjacent to x_1; moreover $y_1 > x_1 > f(V^q)$. But this is a contradiction. □