Solving Stability Problems on a Superclass of Interval Graphs

Carlo Mannino (mannino@dis.uniroma1.it)
Gianpaolo Oriolo (oriolo@disp.uniroma2.it)

Abstract

We introduce a graph invariant, called thinness, and show that a maximum weighted stable set on a graph $G(V, E)$ with thinness k may be found in $O(\frac{|V|}{k})^k$-time, if a certain representation is given. We show that a graph has thinness 1 if and only if it is an interval graph, while a graph with thinness k is the intersection graph of k-dimensional boxes. We investigate properties of the thinness and discuss relationships between graphs with thinness at most two and other superclasses of interval graphs.

We present real a world application where suitable graphs with small thinness naturally occur: the frequency assignment problem (FAP). We show that an efficient search in exponential neighbourhoods for FAP may be done by our polynomial time algorithm. This led us to improving the best known solutions for some benchmark instances of FAP in GSM networks. We discuss other applications where a same approach seems quite natural, among which the single machine scheduling problem. We leave some open questions.

Keywords: Interval Graphs, Stable Sets, Boxicity, Frequency Assignment.

1 Introduction

Interval graphs were introduced by Hajos [14] in 1957 and since then they have been widely studied. In fact, there are many applications, among them scheduling, computational biology, where interval graphs (with possible side constraints) occur. Moreover, they can be recognized in linear time and many classical optimization problems are easy on interval graphs. Good sources for this theory are [8] and [12].

Interval graphs are so powerful and nice (in terms of easiness of optimization) that several generalizations have been proposed. Some generalizations are quite "natural": e.g. intersection graphs of boxes (intervals with higher dimension); others are more "algebraic" and related to characterizations of interval graphs: e.g. AT-free graphs.

Not surprisingly, natural generalizations are often more useful for modeling real world problems. For instance, intersection graphs of two-dimensional boxes (rectangles) occur in the automatic label placement problem [9] and fleet maintenance [20]. Unfortunately, recognizing such graphs is NP-hard and, even worse, some classical optimization problems, such as the maximum stable set problem, are hard to solve even if the box representation is given from the outset [10, 17].

In this paper we introduce a superclass of interval graphs where the maximum (weighted) stable set problem is easy to solve when a certain representation is given from the outset.

We observe that 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 any triple (r, s, t) such that $r < s < t$, if $v_t v_r \in E$, then $v_t v_s \in E$ (Theorem 2.2). Moreover, if we are given for a (interval) graph such an ordering, an $O(|V|)$-time dynamic programming algorithm finds a maximum weighted stable set.

We generalize the previous argument. Namely, we define a graph $G(V, E)$ to be *k-thin* if we can define for V an ordering $\{v_1, \ldots, v_n\}$ and a partition into k classes $V^1, \ldots, V^k$ such that, for any 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. We show that a maximum weighted stable set S^* on a k-thin graph may be found in $O(\frac{|V|}{k})^k$-time, if the ordering and the partition are given from the outset (Theorem 3.1); we enhance this result for a subclass of k-thin graphs, called regular k-thin graphs, which is relevant in several applications (Theorem 3.3). We obtain similar results for the case where the (maximum weighted) stable set S^* has to satisfy additional constraints: namely, for any h, the size $|S^* \cap V^h|$ is given (Theorem 3.4 and Theorem 3.5) .

We define the *thinness*, $thin(G)$, of a graph G as the minimum k for which G is k-thin (therefore a graph has thinness 1 if and only it is interval). We discuss bounds on $thin(G)$ and show that a graph with thinness k is the intersection graph of k-dimensional boxes (Theorem 4.5). We investigate relationships between graphs with thinness at most two and other superclasses of interval graphs: co-comparability and AT-free graphs.

Even if the definition of k-thin graphs is related to an algebraic characterization of interval graphs, we discuss some real world applications where suitable graphs with small thinness naturally occur. We mainly concentrate on the frequency assignment problem in GSM networks [1]. We show that an efficient search in exponential neighbourhoods for this problem may be done by finding a suitable stable set on a certain graph and that this can be done by our polynomial time algorithm. This led us to improving the best known solutions for the Siemens instances of the COST259 [7] test bed for the frequency assignment problem. We show that a similar approach can be taken for solving the problem when frequency hopping is allowed.

We eventually outline another application where solving stable set problems on k-thin graphs is relevant, the single machine scheduling problem. We close by addressing some open questions.

The paper is organized as follows. We close this section with some definitions. In Section 2 we review interval graphs and a dynamic programming algorithm for the maximum (weighted) stable set problem on interval graphs. In Section 3 we introduce k-thin graphs and extend the former algorithm. Section 4 is dedicated to investigating additional properties of the thinness. In Section 5 we describe applications to frequency assignment in GSM networks. In Section 6 we deal with a scheduling problem. Finally, some open questions are discussed in Section 7.

In the sequel of the paper we will use the standard graph theoretical notation. A graph will always be a finite, undirected and loopless graph. We respectively 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.

A *stable set* $S \subseteq V$ is a set of mutually non-adjacent vertices; we denote by $\alpha(G)$ the maximum size of a stable set of G. We often suppose that, for each vertex $v \in V$, a weight $w(v) \in \mathcal{R}_+$ is given. In this case, we denote by $\alpha_w(G)$ the maximum weight of a stable set of G (the weight of a stable set is defined as the sum of the weights of its elements).

For each vertex v, we denote by $N(v)$ the *neighbourhood* of v, that is the set of vertices which v is adjacent to. Three vertices u, v, w of G form an *asteroidal triple* of G if for every pair of them there is a path connecting the two vertices that avoids the neighbourhood of the remaining vertex. G is *AT-free* if no three vertices form an AT. We denote by C_n the *hole* with n vertices, that is $V(C_n) = \{v_1, \ldots, v_n\}$ and $E(C_n) = \{v_i v_{i+1} : 1 \leq i \leq n\}$ (sums are taken modulo n).

A graph is a *comparability* graph if it has a transitive orientation. A graph is a *co-comparability* graph if it is the complement of a comparability graph.

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.

We often deal with orderings on the set V of vertices of a graph. In particular, all the orderings we consider are *total*, but for sake of shortness, we refer them just as *ordering*. When there is no risk of confusion, we simply represent an ordering $>$ by writing V as $\{v_1, v_2, \ldots, v_n\}$, meaning that $v_j > v_i$ if and only if $j > i$.

2 Stable set problems on interval graphs

Perhaps the best known example of intersection graphs is the interval graph. A graph $G(V, E)$ is an *interval graph* if it is the intersection graph of a set of intervals of a linearly ordered set (like the real line). In the following, we refer to such representation as the *interval representation*; in other words, a graph is an interval graph if and only if it admits an interval representation.

There are several alternative characterizations of interval graphs. Probably, the most famous is due to Gilmore and Hoffman.

Theorem 2.1 **[11]**

(i) G *is an interval graph;*

(ii) G *contains no induced* C_4 *and is a co-comparability graph;*

(iii) the maximal cliques of G *can be linearly ordered such that for each vertex* v*, the maximal cliques containing* v *occur consecutively.*

An interval graph $G(V, E)$ may be recognized - and an interval representation may be built - in time $O(|V| + |E|)$ [4]. Also many classical optimization problems are easy on interval graphs; for instance, starting from an interval representation, a simple algorithm finds a maximum (weighted) clique in $O(|V| \log |V|)$-time. It is less trivial that, starting from an interval representation, also a maximum weighted stable set on an interval graph can be found in $O(|V| \log |V|)$-time: this was first observed in [16]. From our point of view, that is related to the following theorem which gives another (to the best of our knowledge not stated before) definition of interval graphs.

Theorem 2.2 *A graph* $G(V, E)$ *is an interval graph if and only if there exists an ordering* $\{v_1, v_2, \dots, v_n\}$ *of* V *such that, for any triple* (r, s, t)*,* $1 \leq r < s < t \leq n$*, if* $v_t v_r \in E$*, then* $v_t v_s \in E$*.*

Proof. Sufficiency. Associate with every v_i the interval $I_i = [a_i, b_i]$, where $b_i = 2i + 1$ and $a_i = 2j$, where j is the smallest index such that v_j is in the set $v_i \cup N(v_i)$ (hence, $a_i \leq 2i$). Let $\bar{G}$ be the interval graph associated with intervals $I_1, \dots, I_n$. We claim that $\bar{G} = G$. Suppose the contrary. Then there exist two vertices v_r, v_s with $s > r$ such that either i) $v_r v_s \in E$, but I_r and I_s do not intersect or ii) $v_r v_s \notin E$, but I_r and I_s do intersect.

Case i). We have $b_s = 2s + 1 > 2r + 1 = b_r$. Since $v_r \in N(v_s)$ and $r < s$ we have $a_s \leq 2r$. But the I_s and I_r intersect, a contradiction.

Case ii) Let q be the smallest index such that v_q is in the set $v_s \cup N(v_s)$. Clearly $q \geq r + 1$: otherwise, there are three vertices v_s, v_r, v_q with $s > r > q$ such that $v_s v_q \in E$ but $v_s v_r \notin E$. But then $a_s = 2q \geq 2(r + 1) > 2r + 1 = b_r$ and I_s and I_r do not intersect, a contradiction.

Necessity. Let $G(V, E)$ be an interval graph and let $\{v_1, \dots, v_n\}$ be any ordering of the vertices such that, denoting by $I_i = [a_i, b_i]$ the interval associated with $v_i \in V$, it is $b_i \geq b_{i-1}$ for $i = 2, \dots, n$. Suppose that there exist $t > s > r$ such that $v_t v_r \in E$ but $v_t v_s \notin E$: $v_t v_r \in E$ implies that $a_t \leq b_r$. But then, $b_r \leq b_s \leq b_t$ implies that $b_s \in [a_t, b_t]$, which in turn implies that I_s and I_t intersect and that $v_s v_t \in E$, a contradiction. □

Starting from an interval representation of a graph, an ordering satisfying Theorem 2.2 may be built in $O(|V| \log |V|)$-time, since we just need to order intervals by increasing right endpoints. If we are given such an ordering, a maximum weighted stable set on G may be may be found in $O(V)$ time via dynamic programming: this is crucial for the rest of the paper so we will go into details. A similar, but somehow more involved, argument was used in [16].

We start with an useful lemma.

Lemma 2.3 *Let $G(V,E)$ be a graph, $X = \{v_1, \dots, v_p\}$ a subset of V, $p \geq 2$, and $v \in V \setminus X$. The following statements are equivalent:*

(i) for any pair (i,j), $1 \leq i < j \leq p$, if $vv_i \in E$ then $vv_j \in E$;

(ii) either $\{v_1, v_2, \dots, v_p\} \cap N(v) = \emptyset$ or there exists r, $1 \leq r \leq p$, such that $\{v_1, v_2, \dots, v_p\} \cap N(v) = \{v_r, v_{r+1}, \dots, v_p\}$.

Proof. (ii) $->$ (i): trivial. (i) $->$ (ii): suppose that $\{v_1, v_2, \dots, v_p\} \cap N(v) \neq \emptyset$ and let $r \leq p$ be the smallest index such that $v_r v \in E$. If $\{v_1, v_2, \dots, v_p\} \cap N(v) \neq \{v_r, v_{r+1}, \dots, v_p\}$, then there would exist s such that $r < s \leq p$ and $vv_r \in E$ while $vv_s \notin E$. $\square$

Hence, suppose that we are given an ordering $\{v_1, v_2, \dots, v_n\}$, satisfying Theorem 2.2, for the vertices of an (interval) graph $G(V,E)$. Let w_i be the weight of each vertex v_i and, for any $j \leq n$, denote by $\alpha_w(j)$ the maximum weight of a stable set S such that $S \subseteq \{v_1, \dots, v_j\}$; finally set $\alpha_w(0) = 0$.

Let $1 \leq j \leq n$; from Lemma 2.3 it follows that either $\{v_1, \dots, v_{j-1}\} \cap N(v_j) = \emptyset$, or there exists $i < j$ such that $\{v_1, \dots, v_{j-1}\} \cap N(v_j) = \{v_i, v_{i+1}, \dots, v_{j-1}\}$. In the former case, let $b(j) = j-1$; in the latter case, let $b(j) = i-1$: in any case, $b(j)$ will be equal to the largest index $l < j$ such that $v_l v_j \notin E$ ($b(j) = 0$ if $v_l v_j \in E$ for any $1 \leq l < j$). Therefore, the following dynamic programming equation holds:

$$\alpha_w(j) = \max \begin{cases} \alpha_w(j-1) \\ \alpha_w(b(j)) + w_j \end{cases} \tag{1}$$

and $\alpha_w(n) = \alpha_w(G)$ may be found in $O(n)$-time. Namely, if we denote by $G_j = G[\{v_1, \dots, v_j\}]$, then $\alpha_w(j) = \alpha_w(G_j)$ and the two terms of the above recursion correspond to i) v_j not belonging to or ii) v_j belonging to a maximum stable set S_j^* of G_j, respectively. In fact, i) if v_j does not belong to S_j^*, then S_j^* is an optimum stable set of G_{j-1}, otherwise ii) $v_j \in S_j^*$, and $S^* \setminus \{v_j\}$ is an optimum stable set of $G[V \setminus (\{v_j\} \cup N(v_j))] = G[\{v_1, \dots, v_{b(j)}\}] = G_{b(j)}$.

In Figure 1.a we show an interval graph along with an ordering of the vertices as in Lemma 2.3. In Figure 1.b. and Figure 1.c we show the graphs corresponding to the two terms in the recursive expression for $\alpha(6) = \alpha(G_6)$.

3 Stable set problems on k-thin graphs

Several superclasses of interval graphs have been proposed in literature, and some of them are reviewed in Section 4; in the following we introduce a new one based on ideas from the

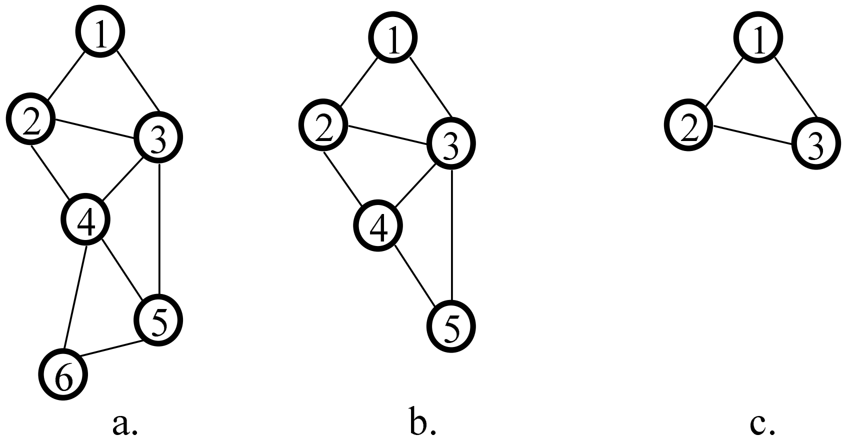

Figure 1: a. $G = G_6$, b. G_5, c. G_3

previous section. This superclass has the feature of allowing easy optimization for the stable set problem when a certain representation is given.

Let $G(V, E)$ be a graph and suppose that we are given for G an ordering $\{v_1, v_2, \ldots, v_n\}$ of V and a partition of V into k classes $V^1, \ldots, V^k$. We say that the partition $(V^1, \ldots, V^k)$ and the ordering $\{v_1, v_2, \ldots, v_n\}$ are *consistent* if, for any triple (r, s, t) such that $1 \leq r < s < t \leq n$, if $v_t v_r \in E$ and v_r and v_s belong to the same class, then $v_t v_s \in E$.

In Figure 2.b and 2.c we show two distinct 2-partitions along with a corresponding consistent ordering for the graph in Figure 2.a.

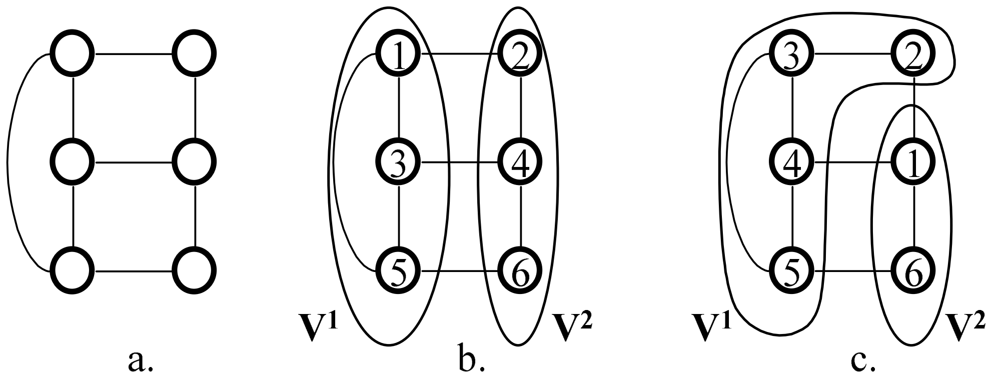

Figure 2: Different partitions and orderings of a graph

As we show in the following, if we are given for a graph $G(V, E)$ an ordering $\{v_1, v_2, \ldots, v_n\}$ and a partition into k classes $V^1, \ldots, V^k$ which are consistent, then a maximum weighted stable set on G may be found in $O(\frac{|V|}{k})^k$-time by dynamic programming.

Let p^h be the size of each class V^h: therefore $\sum_{h=1..k} p^h = n$. Without loss of generality, we assume that each class V^h is non-empty, i.e. $p^h > 0$ for any h (otherwise the following applies for $k = k - 1$). Also we assume on each V^h the ordering induced by the ordering on V and write $V^h = \{v_1^h, \ldots, v_{p^h}^h\}$. Finally, for any $1 \le i \le n$, let $V_i^h = V^h \cap \{v_1, \ldots, v_i\}$.

Let $\mathcal{K}$ be the set of all the k-uplas $(j^1, \ldots, j^k)$ such that $0 \le j^h \le p^h$. For any $(j^1, \ldots, j^k) \in \mathcal{K}$, we denote by $\alpha_w(j^1, ..., j^k)$ the maximum weight of a stable set S such that, for any $1 \le h \le k$, $S \cap V^h \subseteq \{v_1^h, \ldots, v_{j^h}^h\}$: in particular, $S \cap V^h = \emptyset$ if $j^h = 0$. Observe that $\alpha_w(G) = \alpha_w(p^1, \ldots, p^k)$.

Let $(j^1, \ldots, j^k) \neq (0, \ldots, 0)$ be a k-upla of $\mathcal{K}$. We want to evaluate $\alpha_w(j^1, \ldots, j^k)$. Consider the (non-empty) set of vertices $\{v_{j^h}^h, 1 \le h \le k : j^h > 0\}$ and let f be such that $v_{j^f}^f$ is the highest element of this set with respect to the ordering $>$ (that is, for any $h \neq f : j^h > 0$, $v_{j^f}^f > v_{j^h}^h$). Let $1 \le i \le n$ be such that $v_{j^f}^f = v_i$.

For any $h \neq f$, define $X^h = \{v_1^h, \ldots, v_{j^h}^h\}$ and define $X^f = \{v_1^f, \ldots, v_{j^f-1}^f\}$.

Claim. For each h, there exists $0 \le b^h \le |X^h|$ such that $X^h \cap N(v_i) = \{v_{b^h+1}^h, v_{b^h+2}^h, \ldots, v_{|X^h|}^h\}$. In particular, $b^h = |X^h|$ if $X^h \cap N(v_i) = \emptyset$ and $b^h = 0$ if $X^h \subseteq N(v_i)$.

Proof. First, observe that $X^h \subseteq V_{i-1}^h$ (in particular $X^f = V_{i-1}^f$). Then recall that, since the ordering and the partition are consistent, from Lemma 2.3 either $V_{i-1}^h \cap N(v_i) = \emptyset$ or there exists g^h such that $V_{i-1}^h \cap N(v_i) = \{v_{g^h}^h, v_{g^h+1}^h, \ldots, \ldots, v_{V_{i-1}^h}^h\}$. In the former case, $X^h \cap N(v_i) = \emptyset$; in the latter case $X^h \cap N(v_i) = \emptyset$ if $g^h > |X^h|$, $X^h \cap N(v_i) = \{v_{g^h}^h, v_{g^h+1}^h, \ldots, v_{|X^h|}^h\}$ if $g^h \le |X^h|$.

An illustration of the above claim is given in Figure 3.

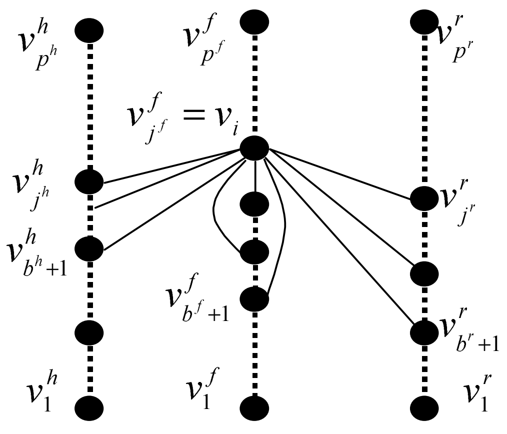

Figure 3: An illustration of the claim

Therefore, the following dynamic programming equation holds:

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

having set $\alpha_w(0,...,0) = 0$.

Observe that, in order to solve the previous equation and evaluate $\alpha_w(G) = \alpha_w(p^1,\ldots,p^k)$, we need an ordering on $\mathcal{K}$ such that, if we evaluate $\alpha(\cdot)$ according to this ordering, when we evaluate $\alpha_w(j^1,...,j^k)$, we have *already* evaluated $\alpha_w(j^1,\ldots,j^{f-1},j^f-1,j^{f+1},\ldots,j^k)$ and $\alpha_w(b^1,...,b^k)$. Since, for any h, $b^h \leq j^h$ and $b^f < j^f$, it is easy to see that such an ordering is *any* linear extension of the following partial order $>_{\mathcal{K}}$ defined on $\mathcal{K}$: $(j^1,..,j^k) >_{\mathcal{K}} (i^1,..,i^k)$ if and only if $\sum_h j^h > \sum_h i^h$.

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

We say that a graph $G(V,E)$ is *k-thin* if there exists an ordering $\{v_1,v_2,\ldots,v_n\}$ of V and a partition of V into k classes which are consistent. Observe that the previous definition allows for some classes of the partition to be empty: in this case the graph is $(k-1)$-thin too. We resume our results in the next theorem.

Theorem 3.1 *Suppose that for a (k-thin) graph G we are given an ordering and a partition into k classes which are consistent. Then a maximum weighted stable set on G may be found in $O(\frac{|V|}{k})^k$-time.*

3.1 Regular k-thin graphs

In this section we deal with a particular class of k-thin graphs which will be exploited to model the relevant applications of Section 5. A graph $G(V,E)$ is a *regular* k-thin graph if:

(i) $V = \{v^h_j, h = 1\ldots k, j = 1\ldots p\}$;

(ii) for each $1 \leq h \leq k$, there exists $\delta^h \geq 1$ such that $\{v^h_1,\ldots,v^h_{i-1}\}\backslash N(v^h_i) = \{v^h_1,\ldots,v^h_{i-\delta^h}\}$($= \emptyset$, if $i-\delta^h \leq 0$).

(iii) for each ordered pair (h,l), $h \neq l$, there exists $\phi^{h,l} \geq 0$ such that $\{v^l_1,\ldots,v^l_i\} \setminus N(v^h_i) = \{v^l_1,\ldots,v^l_{i-\phi^{h,l}}\}$ ($= \emptyset$, if $i-\phi^{h,l} \leq 0$).

Observe that the definition is consistent if, for any pair (h,l): $h \neq l$, $\phi^{h,l} > 0$ if and only if $\phi^{l,h} > 0$. A regular 2-thin graph with $\delta^1 = 3$; $\delta^2 = 2$; $\phi^{1,2} = 1$; $\phi^{2,1} = 2$ is shown in Figure 4.

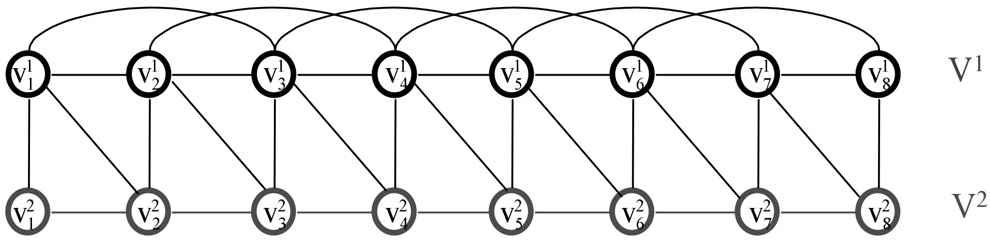

Figure 4: A regular 2-thin graph

If G satisfies (i)-(iii), then it is k-thin, since the ordering $\{v_1^1, \ldots, v_1^k, \ldots, v_p^1, \ldots, v_p^k\}$ and the partition $(V^1, \ldots, V^k)$, where $V^i = \{v_1^i, \ldots, v_p^i\}$, are consistent. Define $\rho(G) = \max_{h,l} |\delta^h - \phi^{h,l}|$; we have the following lemma.

Lemma 3.2 *Let $G(V,E)$ be a regular k-thin graph with $\rho(G) = 0$. Then G is an interval graph.*

Proof. Consider the ordering $\{v_1^1, \ldots, v_1^k, \ldots, v_p^1, \ldots, v_p^k\}$ and a vertex v_i^h. Observe that $\{v_1^1, \ldots, v_1^k, v_2^1, \ldots, v_{i-2}^h, v_{i-1}^h\} \setminus N(v_i^h) = \{v_1^1, \ldots, v_1^k, \ldots, v_{i-\delta^h}^1, \ldots, v_{i-\delta^h}^k\}$. Then the ordering is consistent with V. $\square$

From Theorem 3.1 a maximum weighted stable set on a regular k-thin graph G may be found in $O(p^k)$-time (if the ordering and the partition are given). But, as suggested by Gunter Rote, we can lower this complexity down to $O(p \cdot (\rho(G)+1)^k)$-time. In the following, we assume that $\rho(G) \geq 1$ (if $\rho(G) = 0$ the graph is interval and we know that we can lower the complexity down to $O(p \cdot k)$).

The crucial idea is the following. Let $\rho = \rho(G)$. For computing $\alpha_w(G)$, we do not need to evaluate $\alpha(\cdot)$ (see Equation 2) for each k-pla of the set $\mathcal{K} = \{(j^1, \ldots, j^k) : 0 \leq j^h \leq p\}$, but just for k-plas $(j^1, \ldots, j^k)$ such that, for any $h \neq l$, $|j^l - j^h| \leq \rho$. This is shown in the following; for simplicity we deal with the case $k = 2$.

Let $V^1 = \{v_1^1, \ldots, v_p^1\}$ and $V^2 = \{v_1^2, \ldots, v_p^2\}$; we know that the ordering $\{v_1^1, v_1^2, v_2^1, v_2^2, \ldots, v_p^1, v_p^2\}$ is consistent with (V^1, V^2). Observe that it is $v_j^2 > v_i^1$ if $j \geq i$ while $v_i^1 > v_j^2$ if $i > j$. Recall that $\rho = \max(|\delta^1 - \phi^{1,2}|, |\delta^2 - \phi^{2,1}|)$. Then $\alpha_w(G) = \alpha_w(p,p)$ and, if $j \geq i$, we have:

$$\alpha_w(i,j) = \max \begin{cases} \alpha_w(i, j-1) \\ \alpha_w(\min\{i, j - \phi^{2,1}\}, j - \delta^2) + w(v_j^2) \end{cases} \tag{3}$$

while, if $i > j$, we have:

$$\alpha_w(i,j) = \max \begin{cases} \alpha_w(i-1,j) \\ \alpha_w(i-\delta^1, \min\{j, i-\phi^{1,2}\}) + w(v_i^1) \end{cases} \tag{4}$$

We use induction and show that, if for the current pair (i,j) we have that $|j-i| \leq \rho$, then the same holds for the two terms arising either from (3) or from (4). Observe that, trivially, $|p-p| \leq \rho$.

Without loss of generality we assume that $j \geq i$. By induction, we may assume that $j - i \leq \rho$.

Let $\alpha_w(i_1, j_1) = \alpha_w(i, j-1)$ be the first term of the recursive equation (3). If $j = i$ then $|j_1 - i_1| = 1$. Otherwise $j > i$ and then $|j_1 - i_1| = j - i - 1 < j - i = |j - i|$.

Let $\alpha_w(i_2, j_2) = \alpha_w(\min\{i, j - \phi^{2,1}\}, j - \delta^2)$ be the second term of the recursive equation (3). First, suppose $\min\{i, j-\phi^{2,1}\} = i$, that is $j - i \geq \phi^{2,1}$. In this case, $|j_2 - i_2| = |j - \delta^2 - i|$. If $i \leq j - \delta^2$, then $|j - \delta^2 - i| = j - i - \delta^2 \leq j - i = |j - i|$. Else, if $i > j - \delta^2$, then $|j - \delta^2 - i| = \delta^2 - (j - i) \leq \delta^2 - \phi^{2,1} \leq \rho$.

Finally, when $\min\{i, j - \phi^{2,1}\} = j - \phi^{2,1}$ we have $|j_2 - i_2| = |(j - \delta^2) - (j - \phi^{2,1})| = |\delta^2 - \phi^{2,1}| \leq \rho$.

A similar results hold for the case where $k > 2$. Therefore, we have the following theorem.

Theorem 3.3 (MANNINO, ORIOLO, ROTE) *Let $G(V, E)$ be a regular k-thin graph for which we are given an ordering and a partition satisfying $(i)-(iii)$, $\rho(G) \geq 1$. A maximum weighted stable set on G may be found in $O(p \cdot (\rho(G) + 1)^k)$-time.*

3.2 Stable set with side constraints

In this section we deal with the problem of finding a stable set with maximum weight on a k-thin graph when some *side constraints* are to be met. Namely, we consider the following problem.

We are given a (k-thin) graph $G(V, E)$, together with an ordering and partition of V into k classes V^1, .. , V^k which are consistent, and, for each class V^h, a positive integer d_h. We want to find a stable set S with maximum weight such that $|\, S \cap V^h \,| = d_h$ for $h = 1, \ldots, k$.

Relevant applications of this problem will be discussed in Section 5. In the following, we show how to generalize equation (1) to this case.

Let $V^h = \{v_1^h, \ldots, v_{p^h}^h\}$; denote as $\mathcal{K}$ the set of all the k-uplas $(j^1, \ldots, j^k)$ such that $0 \leq j^h \leq p^h$ and as $\mathcal{D}$ the set of all the k-uplas $(r^1, \ldots, r^k)$ such that $0 \leq r^h \leq d_h$. Finally, for any $(j^1, \ldots, j^k) \in \mathcal{K}$ and $(r^1, \ldots, r^k) \in \mathcal{D}$, denote as $\alpha_w(j^1, ..., j^k; r^1, \ldots, r^k)$ the maximum weight of a stable set S such that, for any $1 \leq h \leq k$, $|\, S \cap V^h \,| = r^h$ and $S \cap V^h \subseteq \{v_1^h, \ldots, v_{j^h}^h\}$.

Let $(j^1,..,j^k) \neq (0,..,0) \in \mathcal{K}$ and $(r^1,..,r^k) \in \mathcal{D}$. If we define j^f and the vector b as for equation (1) (and assume without loss of generality that $r^f > 0$), then the following dynamic programming equation holds:

$$\alpha_w(j^1,...,j^k;r^1,...,r^k) = \max \begin{cases} \alpha_w(j^1,..,j^{f-1},j^f-1,j^{f+1},..,j^k;r^1,...,r^f,..,r^k) \\ \alpha_w(b^1,..,b^k;r^1,..,r^{f-1},r^f-1,r^{f+1},..,r^k) + w(v^f_{j^f}) \end{cases} \quad (5)$$

Hence, by the same arguments of Theorem 3.1, we have the following theorem.

Theorem 3.4 *Suppose that we are given a (k-thin) graph $G(V,E)$ together with an ordering and a partition of V into k classes $V^1,\ldots,V^k$ which are consistent, and, for each class V^h, an integer d_h. Then a maximum (minimum) weighted stable set S, such that $\mid S \cap V^h \mid = d_h$ for $h = 1,\ldots,k$, may be found in $O((\frac{|V|}{k})^k \cdot d^k)$-time, where $d = \max\{d_h, 1 \leq h \leq k\} + 1$.*

When G is a regular k-thin graph, we have the following straightforward extension of Theorem 3.3:

Theorem 3.5 *Let $G(V,E)$ be a regular k-thin graph for which we are given an ordering and a partition satisfying $(i)-(iii)$ and, for each class V^h, an integer d_h. Then a maximum (minimum) weighted stable set S, such that $\mid S \cap V^h \mid = d_h$ for $h = 1,\ldots,k$, may be found in $O(p(\rho(G)+1)^k d^k)$-time, where $d = \max\{d_h, 1 \leq h \leq k\} + 1$.*

We close this section with a remark on the solution to a related problem.

Partition Stable Set Problem (PSS-problem). Let $G(V,E)$ be a graph, $V^1,\ldots,V^k$ be a partition of V into k classes and $d_1,\ldots,d_k$ a set of positive integers: find a stable set S of maximum weight such that $\mid S \cap V^h \mid = d_h$ for $h = 1,\ldots,k$.

Lemma 3.6 *If $k \geq 2$, the* PSS*-problem on an interval graph may be solved in $O((\frac{|V|}{k})^k \cdot d^k)$-time, where $d = \max\{d_h, 1 \leq h \leq k\} + 1$.*

Proof. Since G is interval, there exists an ordering $\{v_1,\ldots,v_n\}$ which is consistent with V. This ordering may be built in $O(|V|\log|V|)$ (see Theorem 2.2). Observe that this ordering is consistent with the partition $(V^1,\ldots,V^k)$; for, suppose the contrary: then there exist $h \in \{1,\ldots,k\}$ and $r \leq s \leq t$ such that $v_r, v_s \in V^h$, $v_rv_t \in E$, but $v_sv_t \notin E$, a contradiction. The statement follows therefore from Theorem 3.4.

4 Investigating thinness

In the previous section, we have defined a graph $G(V,E)$ to be *k-thin* if there exist an ordering of V and a partition of V into k classes $V^1,\ldots,V^k$ which are consistent and we have proved that, in this case, a maximum weighted stable set for G may be found in $O(\frac{|V|}{k})^k$-time if the ordering and the partition are given from the outset.

We define the *thinness* of a graph $G(V,E)$, denoted by $thin(G)$, as the smallest k such that G is k-thin. A simple argument shows that this definition is consistent since $thin(G) \leq |V|$. Indeed, consider the partition of $V(G)$ into n non-empty classes of size one (that is, each class has exactly one vertex). Any ordering on V is consistent with this partition.

We omit the proof of the following simple corollaries; in particular Corollary 4.2 directly follows from Theorem 2.2.

Corollary 4.1 *If H is an induced subgraph of G, then $thin(H) \leq thin(G)$.*

Corollary 4.2 *The thinness of a graph G is 1 if and only if G is an interval graph.*

Corollary 4.3 *The subgraph induced on a k-thin graph by any class V^h, $1 \leq h \leq k$, is an interval graph.*

Unfortunately, as one would expect, there exist graphs whose thinness is $O(n)$, where n is the size of the graph. Denote by T^n the graph with n vertices which is $(n-2)$-regular (the complement of T^n is a matching of size $\frac{n}{2}$). Trivially, such a graph exists only if n is even.

Theorem 4.4 *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 any 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 each class, denote by $f(V^h)$ the smallest element of V^h with respect to the ordering $>$. Of course, 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. □

Corollary 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 [21]. For a graph G the boxicity $b(G)$ is the minimum dimension d such that G is the intersection graph of boxes in d-dimensional space. A graph with boxicity 2 is shown in Figure 5.

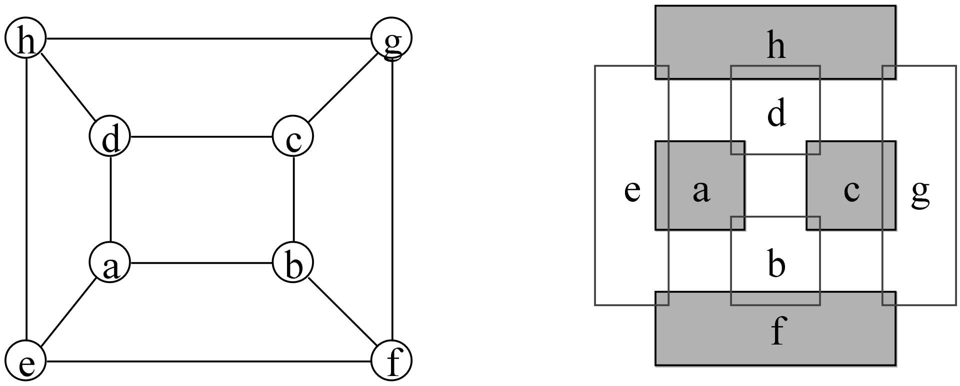

Figure 5: A graph with boxicity 2

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 bounded by its boxicity (and therefore Theorem 4.4 is not surprising, since it is known that there are graphs with n vertices and boxicity $\lfloor \frac{n}{2} \rfloor$ [23, 6]).

For some $k \geq 2$, let $G^1(V, E^1)$, $G^2(V, E^2)$, ... , $G^k(V, E^k)$ be graphs with the same vertex set V. We define *intersection* of G^1, G^2, ... , G^k the graph $G^1 \cap G^2 ... \cap G^k = G(V, E^1 \cap E^2 \cap ... \cap E^k)$. Trivially, if G is the intersection of k interval graphs, then $b(G) \leq k$.

Theorem 4.5 *$thin(G) \geq b(G)$.*

Proof. Let $G(V, E)$ be a graph with thinness k. By definition, there exists an ordering $<$ of V and a partition of V into k classes $V^1, \ldots, V^k$ which are consistent. For each $1 \leq h \leq k$, let $V^h = \{v_1^h, \ldots, v_{p^h}^h\}$ 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 any $u \in V \setminus V^h$ and let r be the smallest index such $uv_r^h \in E$. Then $uv_i^h \in E^h$ for $i = r, \ldots, p^h$.

We claim that $G = I^1 \cap \ldots \cap I^k$ and that each 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 $w_1 = v_1^h$, …, $w_{p^h} = v_{p^h}^h$, 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 4.4, I^h is interval.

We show now that $G = I^1 \cap \ldots \cap I^k$. 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 without loss of generality that $w < u$ in the global ordering and let $w \in V^j$. Recall that $V^j = \{v_1^j, \dots, v_{p^j}^j\}$ and let $w = v_m^j$. Since $uw \in E^j$, by (iii) there exists an index $l \leq m$ such that vertex $uv_l^j \in E$. If $l = m$ then $uv_l^j = uv_m^j = uw \in E$, while if $l < m$ then $v_l^j < v_m^j = w < u$ and consistency implies $uw \in E$. $\square$

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. A simple but tedious proof (which we omit) shows that the graph in Fig. 5 has boxicity 2 and thinness 3.

Again, this is not surprising, because while we have shown that a maximum weighted stable set for a k-thin graph (with fixed k) may be found in poly-time if a consistent ordering and a partition are given, this is not the case dealing with boxicity. In fact, finding a maximum stable set on a graph with boxicity 2 is hard even if the box representation is given [10, 17].

We close this section by discussing relations between 2-thin graphs and other generalizations of interval graphs. Recall that interval class are a subclass of co-comparability graphs (Theorem 2.1). Moreover Golumbic, Monma and Trotter proved that every co-comparability graph is AT-free [13], that is co-comparability graphs are a subclass of AT-free graphs. Therefore, AT-free graphs, as well as co-comparability graphs and 2-thin graphs, generalize interval graphs. One might wonder if every co-comparability graphs is a 2-thin graph, or vice versa if every 2-thin graph is AT-free. In both cases, the answer is negative.

First, consider the graph T^6 defined above. $\bar{T}^6$ is trivially a comparability graph. On the other hand, we know from Theorem 4.4 that T^6 is not a 2-thin graph.

Then, consider any hole C_k, $k \geq 6$. C_k is a 2-thin graph. In fact, suppose that $V(C_k) = \{v_1, \dots, v_k\}$; let $V^1 = \{v_1\}$, $V^2 = V \setminus v_1$ and let $>$ be such that $v_j > v_i$ if and only if $j > i$.

Then the ordering $>$ and the partition (V^1, V^2) are consistent. On the other hand, it is easy to check that C_k is not AT-free.

Figure 6 summarizes the previous discussion.

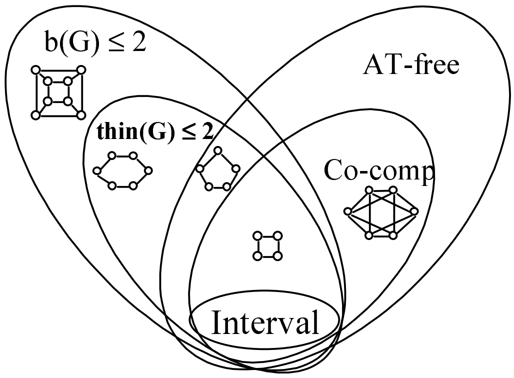

Figure 6: Different superclasses of interval graphs

5 The frequency assignment problem

The Frequency Assignment Problem (FAP) is the problem of assigning a limited number of radio frequencies to a network of transmitters with interference limitations (for a survey see [1, 18]). In particular, every transmitter v of a network T requires a given number d_v of transmission frequencies selected from a set $F = \{1, \ldots, f_{max}\}$ of available frequencies. Close transmitters can interfere, resulting in a deteriorated reception. However, by letting the transmission frequencies be "far" enough in the spectrum, interference may be cut down.

For our purposes, an instance of FAP will be described by means of the 4-pla (T, C, d, F), where T is a set of transmitters, $F = \{1, \ldots, f_{max}\}$ is a set of available frequencies, $C \in Z_+^{|T|\times|T|}$ is a *distance requirements* matrix and $d \in Z_+^{|T|}$ is a *demand* vector.

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

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

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

Observe that C is symmetric, $C_{uv} \geq 0$ for each pair uv and without loss of generality $C_{uu} \geq 1$ for each u. In general, we want to solve the following problem. Let (T, C, d, F) be an instance of FAP: does it admit a feasible frequency assignment?

We show how to solve the above problem by finding a suitable stable set in a graph. The graph $G(W, E)$ is defined as follows:

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

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

Graph G is called the *conflict graph* associated with the instance (T, C, d, F). Every stable set of G corresponds to an assignment of frequencies to the transmitters in T which does not violate any distance constraint (and vice versa). So, an instance (T, C, d, F) of FAP admits a feasible frequency assignment if and only if there exists a stable set S of G such that $|S \cap W_v| = d_v$ for each $v \in T$, where, for each $v \in T$, we denote as $W_v = \{(v, f) : f \in F\}$. The conflict graph has a structure we are familiar with.

Define on the set W the ordering $\{(v_1, 1), \ldots, (v_n, 1), \ldots, (v_1, f_{max}), \ldots, (v_n, f_{max})\}$ and the partition $\{W_{v_1}, \ldots, W_{v_n}\}$; finally set, for each $u \neq v$, $\phi^{u,v} = \phi^{v,u} = C_{uv}$ and $\delta^u = C_{uu}$.

Corollary 5.1 *If $|T| = n$, then G is a regular n-thin graph.*

Corollary 5.2 *For each $Z \subseteq T$, define the subgraph $G_Z = G[\bigcup_{v \in Z} W_v]$. G_Z is a regular $|Z|$-thin graph.*

In practice, large real-life instances are infeasible, i.e. every frequency assignment violates a number of distance constraints. Practitioners face this problem by relaxing distance constraints between *some* pairs of transmitters: this results in a limited interference.

Distance constraints which cannot be relaxed (violated) are called *hard constraints*. In general they stand for some type of hardware limitations. For example, in a mobile network system all of the frequencies assigned to a same transmitter (*base station* or *cell*) u, for which $d_u > 1$, must satisfy a hard distance constraint (typically, $C_{uu} = 3$ for all $u \in T$). Also, when a cluster $Z \subseteq T$ of transmitters is mounted on a same physical support (*site*), all pairs of transmitters in Z must satisfy a hard distance constraint (typically, $C_{uv} = 2$ for any $u, v \in Z$). Additional hard constraints may derive from mobility requirements (*handover*).

As for non-hard constraints, when a frequency g assigned to u and a frequency f assigned to v violate $|g - f| \geq C_{uv}$, we incur a penalty cost $p(u, v, |g - f|)$ which is (somehow) proportional to the corresponding interference level. Therefore, from now on the optimization problem is therefore that of finding a frequency assignment satisfying demand constraints and hard distance constraints which minimizes the sum of penalty costs.

5.1 The extension problem

Extensive experimental experience has shown that, for large scale real-life instances of FAP, local search is by all means the most successful technique to finding good quality frequency assignments (see, for example, the reports on benchmark instances in [7]). The definition of suitable neighbourhoods of a solution is critical in determining the quality of the search. A possible line of attack is to define suitable "large" neighbourhoods, but "easy" to optimize: this technique is generally known as *large-scale neighbourhood search* (see, for example [2]). The rationale is that optimizing in larger neighbourhoods can drastically enhance the search, provided that this can be carried out efficiently.

Coming back to FAP, in [15] Heller and Hellebrandt provide a dynamic programming procedure for efficiently optimizing over the neighbourhood $N_1(s)$ of a solution s, defined as the set of assignments obtained from s by (i) selecting exactly one transmitter v of T and (ii) substituting *all* of the frequencies assigned to v: by this technique they were able to significantly improve the quality of the assignments produced by Simulated Annealing over the COST259 test bed [7, 15]. In the sequel, we show how to generalize this result to larger neighbourhoods.

Namely, we choose a *hard set* Z, i.e. a set of transmitters pairwise hard constrained. Then, for a given solution s, we define $N_{|Z|}(s)$ as the set of assignments obtained from s by substituting *all* of the frequencies assigned to *all* of the transmitters in Z. We show that the optimization task can be carried out efficiently by solving a stable set problem on a regular $|Z|$-thin graph (when $|Z| = 1$, which is the case considered by Heller and Hellebrandt, such graph will be an interval graph). In particular, this procedure provides a large-scale neighbourhood search: while $|N_z(s)| = \prod_{v \in Z}(f_{max})^{d_v}$, the optimization task may be performed in $O(f_{max} \cdot (\rho+1)^{|Z|} \cdot \prod_{v \in Z}(d_v+1))$-time, where ρ is typically small (usually, 1 or 2).

So let Z be a hard set and suppose that we are given a frequency assignment for the other transmitters of the network: that is, for each $v \in T \setminus Z$, we are given a set $F_v \subseteq F : | F_v | = d_v$. We want to extend this assignment by defining an assignment for the transmitters in Z which (j) satisfies all the hard constraints; (jj) minimizes the penalty costs arising from non-hard distance constraints: we call this problem the *extension problem* applied to Z. Observe that, since Z is a hard set, *additional* penalties costs arise only from interference between a transmitter $u \in Z$ and a transmitter $v \notin Z$ (of course there might be penalty costs arising from pairs not in Z). In particular by assigning frequency $g \in F$ to transmitter $u \in Z$, we incur a penalty cost $w_{ug} = \sum_{v \notin Z} \sum_{f \in F_v} p(u, v, |g - f|)$.

The following lemma is a simple consequence of Theorem 3.5.

Lemma 5.3 *Let $Z \subseteq T$ be a hard set, $\rho = \max_{u,v \in Z} |C_{uu} - C_{uv}|$ and $d = \max\{d_v, v \in Z\} + 1$. If $\rho \geq 1$, the extension problem applied to Z can be solved in $O(f_{max} \cdot \max\{\rho+1\}^{|Z|} \cdot d^{|Z|})$-time; if $\rho = 0$ it can be solved in $O(f_{max} \cdot |Z| \cdot d^{|Z|})$-time.*

By exploiting the results of the previous sections, we implemented an efficient local search procedure to the solution of large real-life GSM instances. The search is performed over the neighbourhood $N_{|Z|}(s)$, where Z is any hard set; it turns out that in all these instances $|Z| \leq 9$. All the hard sets are listed and visited in a round robin fashion and the algorithm stops if no improving solution has been found in the last round.

In Table 5.1 we consider four instances of the COST259 test bed [7], the so called *Siemens instances*. The best known solutions to these instances were found by the N_1 neighbourhood search of Heller and Hellebrandt [15]. We were interested in checking if the straightforward application of our extended N_Z neighbourhood search was sufficient for improving such solutions. This was the case, as illustrated in Table 5.1.

Instance	Best Known	N_Z search
Siemens1	23.00	22.96
Siemens2	14.75	14.72
Siemens3	52.55	52.43
Siemens4	80.96	80.80

Table 1: Results of local search over Siemens instances

5.2 The extension problem with frequency hopping

Frequency hopping is a technique which allows to reduce the overall interference by rapidly shifting the transmission frequency assigned to connected mobiles. The frequencies dynamically assigned to a mobile are randomly chosen from a list of frequencies available for the transmitter, the *mobile allocation list* (MAL). In the classical approach to GSM planning, the number of frequencies d_v to be assigned to transmitter v is determined as the minimum number of frequencies which suffices to serve all of the traffic in the transmitter v with high probability (typically 99%). However, frequency hopping schemes benefit from longer lists, because it has been observed that the overall interference may decrease if the number of frequencies assigned to a transmitter increases.

Björklund et al. [3] developed a model which takes this behavior into account. For sake of simplicity, we consider again the *extension problem*, defined in Section 5.1: we must assign a MAL of at least d_v frequencies to a single transmitter v, while keeping fixed the frequencies assigned to the other transmitters. As in the standard FAP, the cost w_{vf} of assigning a single frequency f to a transmitter v may be computed by summing up all the penalties due to non-hard distance constraints violations (see above). Suppose that a MAL $\mathcal{F}_v = \{f_1, \ldots, f_r\}$, $r \geq d_v$, has been assigned to a transmitter v, then the cost $c(\mathcal{F}_v)$ of this MAL is equal to $g(r) \sum_{i=1..r} w_{vf_i}$, where the parameter $g(r)$ depends on the size of the MAL. In particular, there is a trade-off between the value of $g(r)$, defined for r ranging from d_v to some suitable number f_{max}, which decreases as r increases, and the value of sum $\sum_{i=1..r} w_{vf_i}$, which increases as r increases. The STMALP (single transmitter mobile allocation list problem) is that of determining, for a transmitter v, a MAL $\mathcal{F}_v$ minimizing $c(\mathcal{F}_v)$.

In [3] the authors propose a simulated annealing procedure to find a frequency assignment for the entire network: at each iteration a STMALP must be solved to optimize. In order to perform this task in a reasonable amount of time, the authors renounce to the exact solution of STMALP and make use of a simple greedy algorithm. On the other hand, a straightforward amendment of our dynamic equation (5) allows for the efficient solution of STMALP. When dealing with a single transmitter, equation (5) may be rewritten as:

$$\alpha_w(j,r) = \min \begin{cases} \alpha_w(j-1,r) \\ \alpha_w(b,r-1) + w(v_j) \end{cases}$$

where $\alpha_w(j,r)$ denotes the minimum weight of a stable set of size r with vertices in $\{v_1,\ldots,v_j\}$ and $b < r$ is a suitable index. It is easy to check that the following recursive equation solves MALP (for single transmitter):

$$\alpha_w(j,r) = \min \begin{cases} \alpha_w(j-1,r) \\ \alpha_w(b,r-1)\dfrac{g(r)}{g(r-1)} + w(v_j)g(r) \end{cases} \tag{6}$$

Finally, observe that this equation can be extended as to solve for any hard set of transmitters. Moreover, it can be refined by the same arguments of Lemma 5.3.

6 Scheduling on a single machine

We here outline another application where solving stable set problems on k-thin graphs is relevant, the single machine scheduling problem.

Let $J = \{1,2,\ldots,n\}$ be a set of jobs to be scheduled on a single machine and let $P = \{p_1,p_2,\ldots,p_n\}$ be their *processing times*. Let $R = \{r_1,r_2,\ldots,r_j\}$ be the set of the *release dates* when the jobs become available for processing. Let w_j be a weight associated with the job $j \in J$ and let C_j be its completion time in a given schedule. Let $T = \{1,2,\ldots,|T|\}$ denote the planning horizon.

The Single Machine Scheduling Problem with Release Dates (SMSR-problem) consists of scheduling the jobs on a single machine so as to minimize the total weighted completion time of the jobs $\sum_{j\in J} w_j C_j$. The SMSR-problem is known to be NP-hard and has been widely addressed in the literature. A well studied and effective representation for the SMSR-problem is the *time-indexed formulation* (see [22]); basically, this formulation reduces the problem to finding a suitable stable set on a graph $G(W,E)$, defined as follows:

$W = \{(j,t) : j \in J, t \in T\}$: there exists a vertex of G for any pair (j,t) where j is a job and t an instant of the time horizon. Each 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) $f = g$; or (ii) $g > f$ and $g - f < p_v$; or (iii) $f > g$ 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 a job u (job v) started at time g (time f) has not been completed.

For each $j \in J$, let $W_j = \{(j,t) : t \in T\}$.

Lemma 6.1 *G is a regular $|J|$-thin graph with $\rho(G) = 0$ (and therefore is interval).*

Proof. Let $J = \{1, \ldots, |J|\}$. Define on the set W the ordering $\{(1,|T|), \ldots, (|J|,|T|), \ldots, (1,1), \ldots, (|J|,1)\}$ and the partition $\{W_1, \ldots, W_j\}$; finally set, for each ordered pair (i,j), $i \neq j$, $\phi^{i,j} = \delta^i = p_i - 1$. It follows that G is a regular $|J|$-thin graph. On the other hand, since $\rho(G) = 0$, from Lemma 3.2 then G is interval.

Observe that every feasible assignment of jobs to starting times corresponds to a stable set S such that $|S \cap W_j| = 1$ for $j \in J$. In other words, if we give to each 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 a stable set S with minimum weight such that $|S \cap V_j| = 1$ for $j \in J$. The following lemma is a simple consequence of Theorem 3.5.

Lemma 6.2 *The* SMSR*-problem can be solved in $O(|T| \cdot 2^{|J|})$-time.*

7 Open questions

A number of interesting issues concern with the links between thinness and boxicity. Theorem 4.5 shows that the thinness of a graph is less or equal than its boxicity. It would be therefore interesting to characterize graphs for which the boxicity is equal to the thinness and graphs for which the thinness is much larger than the boxicity.

Also, complexity issues are open. We recall that recognizing the boxicity of a graph is NP-hard: this was shown in three steps. First, Cozzens [5] showed that computing the boxicity of a graph is NP-hard; this was improved by Yannakakis [24] to testing whether $b(G) \leq 3$ is NP-complete and by Kratochvil [19] 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.

Turning to practical issues, in Section 5 we have shown that an efficient search can be performed in an exponentially large neighbourhood of the current frequency assignment for GSM networks. Such neighbourhood drastically enlarges the neighbourhood definition proposed by Heller and Hellebrandt [15], which played a crucial role to finding good quality solutions for a number of real-life instances. Following the same line of attack, our neighbourhood search could be embedded in a sophisticated heuristic scheme in order to produce good solutions to large instances of FAP. A similar approach may be applied to the solution of the single machine scheduling problem.

Acknowledgments: Antonio Sassano was the inspiration for this work. He sparkled our interest in the problem, gave us useful hints and made several nice remarks. Definitely, we are in debt with him.

We are also grateful to Rote (see Theorem 3.3) and Amish Waterer for insights during discussion at the Aussois 2002 Workshop in Combinatorial Optimization. We are also grateful to Egon Balas for his kind encouragement.

References

[1] K.I. Aardal, S. van Hoesel, A. Koster, C.Mannino, A. Sassano, *Models and Solutions Techniques for Frequency Assignment Problems* ZIB Report 01-40, Konrad-Zuse-Zentrum, Berlin 2001.

[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] P. Björklund, P. Värbrand and D. Yuan, *An optimization approach for frequency planning in mobile networks with frequency hopping*, Technical Report, Department of Science and Technology, Linköping Insitute of Technology, 2002.

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

[6] 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.

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

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

[9] 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.

[10] 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.

[11] P.C. Gilmore, A.J. Hohhman, *A characterization of comparability graphs and of interval graphs*, Canadian Journal of Mathematics. 16, 1964, 539-548.

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

[13] M.C. Golumbic, C.L. Monma, W.T. Trotter, *Tolerance graphs*, Discrete Applied Mathematics 9, 1984, 157-170.

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

[15] M. Hellebrandt and H. Heller, *A new heuristic method for frequency assignment*, Tech. Report TD(00)003, COST259, Valencia, Spain, 2000.

[16] J.Y. Hsiao, C.S. Tang, and R.S. Chang, *An efficient algorithm for finding a maximum weight 2-independent set on interval graphs*, Information Processing Letters, 43(5), 1992, 229-235.

[17] 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.

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

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

[20] 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.

[21] 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.

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

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

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