Available online at www.sciencedirect.com
ScienceDirect
Operations Research Letters 35 (2007) 1–9
Operations Research Letters
www.elsevier.com/locate/orl

The stable set problem and the thinness of a graph

Carlo Mannino*, Gianpaolo Oriolo, Federico Ricci, Sunil Chandran

Dip. di Informatica e Sistemistica, Universita di Roma, 'La Spapienza', Via Bounarroti 12, I-00185 Roma, Italy

Received 25 August 2005; accepted 23 January 2006
Available online 29 March 2006

Abstract

We introduce a poly-time algorithm for the maximum weighted stable set problem, when a certain representation is given for a graph. The algorithm generalizes the algorithm for interval graphs and that for graphs with bounded pathwidth. By a suitable application to the frequency assignment problem, we improved several solutions to relevant benchmark instances.

Keywords: Thinness; Stable set; Frequency assignment; Pathwidth

1. Introduction

We introduce a superclass of interval graphs for which the maximum (weighted) stable set problem may be solved by dynamic programming in poly-time when a certain representation is given. The new class is introduced by extending a classical characterization [10,12]: a graph $G(V, E)$ is interval if and only if there exists an ordering $\{v_1, \ldots, v_n\}$ of V such that, for each triple (r, s, t) with $r < s < t$, if $v_t v_r \in E$, then $v_t v_s \in E$: if such an ordering is at hand, an $O(|V|)$-time dynamic programming algorithm finds a maximum weighted stable set [6,7].

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) with $r < s < t$, if v_r, v_s belong to the same class and $v_t v_r \in E$, then $v_t v_s \in E$. If we are given for a graph $G(V, E)$ such an ordering and a partition, then a maximum weighted stable set may be found via dynamic programming in $O(|V| \cdot (\rho + 1)^{k-1})$-time, where ρ is always bounded by the maximum degree of a vertex. Since $\rho = 0$ for interval graphs, this extends the result in [6,7].

We define the thinness of a graph G as the minimum k for which G is k-thin. We relate the thinness to the pathwidth of a graph and show that $thinness(G) \leqslant pathwidth(G) + 1$. Also it turns out that our algorithm for finding a maximum weighted stable set on a k-thin graph is in fact a generalization of the weighted stable set algorithm for bounded pathwidth graphs.

Our main application is discussed in Section 4. The frequency assignment problem (FAP) in GSM networks

* Corresponding author.
E-mail address: mannino@dis.uniroma1.it (C. Mannino).

0167-6377/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2006.01.009

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 stable set problem on a graph G_T; since an ordering and a consistent $|T|$-partition of G_T can be easily built, the problem can then be efficiently solved by our dynamic program (usually, $\rho \leqslant 2$). This allows us to implement an effective *search in exponential neighborhoods* [4]. We encapsulated our methodology into a simulated annealing framework and tested the resulting algorithm on the COST259 test bed [5], which consists of a number of large real-life instances of the FAP in GSM networks. The numerical results are extremely good and we were able to improve the best known solutions on most instances of the test bed.

We deal with simple and undirected graphs. $V(G)$ and $E(G)$ are, respectively, the vertex set and the edge set of G. If W is a subset $W \subseteq V$, $G[W]$ is 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 $v \in V$, $\delta(v)$ is the *degree* of v; $\Delta(G)$ is the maximum degree of a vertex of G. We, respectively, denote by $\alpha(G)$ and $\alpha_w(G)$ the maximum size and weight of a stable set of G ($w : V \to \mathscr{R}_+$ is the weight function).

We denote by $N(X)$ the set of neighbors of $X \subset V$, i.e. $N(X)=\{u \notin X : uv \in E \text{ for some } v \in X\}$, where $N(v)=N(\{v\})$. $\overline{N}(v)$ is the set of vertices different from v and not adjacent to v, i.e. $\overline{N}(v)=V-(N(v)\cup\{v\})$. A *k-partition* of V is a partition of V into k classes $V^1,\ldots,V^k$. An *ordering* $\{v_1,v_2,\ldots,v_n\}$ is a linear ordering of V (therefore, $v_j > v_i$ if and only if $j > i$).

Definition 1.1. 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 $1 \leqslant j \leqslant n$:

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

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. The stable set problem on a superclass of interval graphs

A graph G is an *interval graph* if it is the intersection graph of a set of intervals of the real line. If we are given an interval representation of G, then a maximum weighted stable set can be found in $O(|V|\log|V|)$-time via a dynamic programming algorithm: this was first observed in [6] and later in [7]. In the following we review the latter approach. The first "ingredient" is the following theorem providing a characterization of interval graphs.

Theorem 2.1 (*Olariu [10], Ramalingam and Pandu Rangan [12]*). *A graph $G(V,E)$ is interval 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 $1 \leqslant i < j < k \leqslant n$, if $v_k v_i \in E$, then $v_k v_j \in E$.*

The 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 1.1 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 $1 \leqslant i < j < k \leqslant n$, if $v_k v_i \in E(G)$ then $v_k v_j \in E(G)$;*
(ii) [*Consistency*] *for each $1 \leqslant j \leqslant n$, $\overline{N}(v_j)_< = \{v_1,v_2,\ldots,v_{|\overline{N}(v_j)_<|}\}$.*

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 an ordering $\{v_1,\ldots,v_n\}$ of $V(G)$ which is consistent.*

Let $G=(V,E)$ be a (interval) graph with a consistent ordering $\{v_1,\ldots,v_n\}$ of $V(G)$ and weights

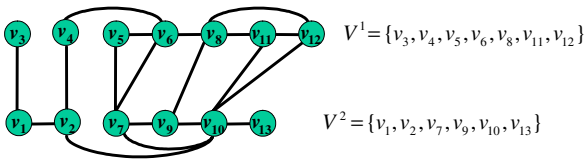

Fig. 1. A graph along with an ordering and a partition of the vertices which are consistent.

$w(v_j)$ for each vertex v_j. Let $\alpha_w(0) = 0$ and, for each $1 \leqslant j \leqslant n$, denote by $\alpha_w(j)$ the maximum weight of a stable set $S_j \subseteq \{v_1, \ldots, v_j\}$. If $v_j \notin S_j$ then S_j is a maximum weighted stable set in $\{v_1, \ldots, v_{j-1}\}$. If $v_j \in S_j$ then $S_j - v_j$ is a maximum weighted stable set in $G[\{v_1, \ldots, v_{j-1}\} - N(v_j)] = G[\overline{N}(v_j)_<]$ by Definition 1.1. Since the ordering is consistent, by Lemma 2.2 $\overline{N}(v_j)_< = \{v_1, v_2, \ldots, v_{|\overline{N}(v_j)_<|}\}$. So, the following recursion correctly evaluates $\alpha_w(j)$ and $\alpha_w(G) = \alpha_w(n)$ may be found in $O(n)$-time:

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

2.1. A superclass of interval graphs

The existence of a consistent ordering is at the basis of the dynamic program (1). On the other hand, there is such an ordering if and only if a graph is interval (Corollary 2.3). We now 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 stable 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 $\{v_1, \ldots, v_n\}$ of V. Suppose that also we are given a *k-partition* $V^1, \ldots, V^k$ of V. Denote by p^h the size of each class V^h and assume that $V^h = \{v_1^h, \ldots, v_{p^h}^h\}$, with $v_1^h < v_2^h < \cdots < v_{p^h}^h$ (i.e. with respect to the ordering). The following lemma is a natural extension of Lemma 2.2. We omit the simple proof.

Lemma 2.4. *Let $G(V, E)$ be a graph with a k-partition $(V^1, \ldots, V^k)$ and an ordering $\{v_1, v_2, \ldots, v_n\}$ of V. The following statements are equivalent*:

(i) *for each triple (i, j, k) such that $1 \leqslant i < j < k \leqslant n$, 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, $\overline{N}(v_j, h)_< = \{v_1^h, v_2^h, \ldots, v_{|\overline{N}(v_j,h)_<|}^h\}$.*

If an ordering and a k-partition are consistent, then the only vertices of a class V^h *lower* than a vertex v_j and *non-adjacent* to it are the first $|\overline{N}(v_j, h)_<|$ ones. An example is in Fig. 1.

Theorem 2.5. *If for a graph G we are given an ordering and a k-partition which are consistent, then a maximum weighted stable set of G may be found in $O((|V|/k)^k)$-time.*

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. Assume that $V^h = \{v_1^h, \ldots, v_{p^h}^h\}$, with $v_1^h < v_2^h < \cdots < v_{p^h}^h$. We use a dynamic program for evaluating $\alpha_w(G)$. Each state of the dynamic program is associated to a k-tuple $(j^1, \ldots, j^k)$ with $0 \leqslant j^h \leqslant p^h$ for each h; the set 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 a stable set $S_{(j^1,\ldots,j^k)} \subseteq \bigcup_{h=1,\ldots,k} \{v_1^h, \ldots, v_{j^h}^h\}$.

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 stable set in $G[\{v_1^1, \ldots, v_{j^1-1}^1\} \cup_{h=2,\ldots,k} \{v_1^h, \ldots, v_{j^h}^h\}]$. In the latter case, $S_{(j^1,\ldots,j^k)} - u$ is also a maximum weighted stable set in $G[\bigcup_{h=1,\ldots,k} \{v_1^h, \ldots, v_{j^h}^h\} - N(u)]$.

Claim. *For* $1 \leqslant h \leqslant p$, $\{v^h_1, \ldots, v^h_{j^h}\} - N(u) = \{v^h_1, \ldots, v^h_{b^h}\}$, *with* $b^h = \min(j^h, |\overline{N}(u,h)_<|)$.

By Lemma 2.4, since the ordering and the partition are consistent, for any class V^h we have that $\overline{N}(u,h)_< = \{v^h_1, v^h_2, \ldots, v^h_{|\overline{N}(u,h)_<|}\}$. Also, since u is the highest vertex in the set $\{v^1_{j^1}, \ldots, v^k_{j^k}\}$, for each h we have that $\{v^h_1, \ldots, v^h_{j^h}\} - N(u) = \{v^h_1, \ldots, v^h_{j^h}\} \cap \overline{N}(u,h)_< = \{v^h_1, v^h_2, \ldots, v^h_{b^h}\}$, where $b^h = \min(j^h, |\overline{N}(u,h)_<|)$.

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

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

Since $b^h \leqslant j^h$ 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 $\mathscr{K}$. 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 $|\mathscr{K}|$ of the set of k-tuples is equal to $(p^1 + 1)(p^2 + 1) \ldots (p^k + 1) \leqslant (|V|/k + 1)^k$, a maximum stable set on G may be found in $O(|V|/k)^k$-time. □

2.1.1. A strengthening of Theorem 2.5

The complexity established by Theorem 2.5 can be refined by observing that indeed only a small subset of k-tuples of $\mathscr{K}$ will be actually enumerated. By simple counting arguments we give an upper bound on the maximum size of this subset. We denote by $\sigma = \{v_1, \ldots, v_n\}$ the ordering and by $\mathscr{V} = (V^1, \ldots, V^k)$ the consistent k-partition.

For each $j \in \{1, \ldots, n\}$, denote as $p(j)$ the highest *index* $i < j$ such that $v_i v_j \notin E$. Recall 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 $p(j) \geqslant |\overline{N}(v_j)_<|$. In Fig. 1, we have, for instance, $p(12) = 9 > |\overline{N}(v_{12})_<| = 8$, $p(10) = 8 > |\overline{N}(v_{10})_<| = 6$ and $p(6) = 3 = |\overline{N}(v_6)_<|$.

Proposition 2.6. *If* $p(j) = |\overline{N}(v_j)_<|$ *for every* j *then* G *is interval.*

Proof. Assume that $\overline{N}(v_j)_< = \{v_{j_1}, \ldots, v_{j_{|\overline{N}(v_j)_<|}}\}$. Since $|\overline{N}(v_j)_<| = p(j)$, then we must have $j_1 = 1$, $j_2 = 2, \ldots, j_{|\overline{N}(v_j)_<|} = p(j)$ and so for each $v_j \in V$, we have that $\overline{N}(v_j)_< = \{v_1, v_2, \ldots, v_{p(j)}\}$. 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. Also let $\rho = \rho(\sigma, \mathscr{V}) = \max_{j,h} \rho(v_j, h)$.

Proposition 2.7. $p(j) - |\overline{N}(v_j)_<| = \sum_h \rho(v_j, h)$.

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

We have two simple corollaries: the former follows from Propositions 2.6 and 2.7; the latter from the definition of ρ and since, for any vertex v and each class V^h, $\rho(v, h) \leqslant \delta(v)$.

Corollary 2.8. *If* $\rho = 0$ *then* G *is interval* (*and vice versa if* G *is interval then* $\rho = 0$ *for any ordering satisfying Lemma* 2.2).

Corollary 2.9. $\rho \leqslant \Delta(G)$.

In 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 $\rho(10, 2) = 2$—in fact, $p(10) = 8$ 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 $\rho = 2$.

It is the size of ρ that really matters for evaluating the complexity of program (2). We have:

Theorem 2.10. *If a consistent ordering* σ *and a* k*-partition* $\mathscr{V}$ *of* $V(G)$ *are given, then a maximum weighted stable set of* G *may be found in* $O(|V(G)| \cdot (\rho(\sigma, \mathscr{V}) + 1)^{k-1})$*-time.*

Proof. We use the same settings as for the proof of Theorem 2.5. Moreover, we define a k-tuple $(j^1, \ldots, j^k)$ to be *relevant* if, for the largest $v^i_{j^i}$ in the k-tuple and any $h \neq i$, the number $n(v^i_{j^i}, v^h_{j^h})$ of vertices in V^h between $v^h_{j^h}$ and $v^i_{j^i}$ is at most ρ, i.e., $n(v^i_{j^i}, v^h_{j^h}) = |\{u \in V^h : v^h_{j^h} < u < v^i_{j^i}\}| \leqslant \rho$. Observe that the starting k-tuple $(p^1, \ldots, p^k)$ of our dynamic programming algorithm (2) is trivially relevant. We then show that, if the program is started at a relevant k-tuple, it only considers relevant k tuples. This implies the statement since, for any vertex v, the number of k-tuples with v largest vertex in the k-tuple is then bounded by $(\rho+1)^{k-1}$.

So suppose by induction that $(j^1, \ldots, j^k)$ is a relevant k-tuple and w.l.o.g. suppose that $v = v^1_{j^1}$ is the largest vertex in the k-tuple. Consider again Eq. (2) and, first, consider the k-tuple $(j^1 - 1, j^2, \ldots, j^k)$. If $v = v^1_{j^1-1}$ is the largest vertex in the k-tuple, then, for $h \neq 1$, $n(v^1_{j^1-1}, v^h_{j^h}) \leqslant n(v^1_{j^1}, v^h_{j^h}) \leqslant \rho$. So suppose w.l.o.g. that $v = v^2_{j^2}$ is the largest vertex in the k-tuple. For $h \neq 1, 2$, $n(v^2_{j^2}, v^h_{j^h}) \leqslant n(v^1_{j^1}, v^h_{j^h}) \leqslant \rho$; $n(v^2_{j^2}, v^1_{j^1-1}) = 0$, since $v = v^1_{j^1} > v^2_{j^2}$.

Consider the k-tuple $(b^1, \ldots, b^k)$. First, suppose that $v^1_{b^1}$ is the largest vertex in the k-tuple, that is, $v^1_{b^1}$ is the highest vertex that is not adjacent to $v = v^1_{j^1}$. Consider a class V^h, $h \neq 1$, such that $b^h < j^h$: by definition, $n(v^1_{b^1}, v^h_{b^h}) = \rho(v^1_{j^1}, h)$. Consider then a class V^h, $h \neq 1$, such that $b^h = j^h$: we have that $n(v^1_{b^1}, v^h_{j^h}) \leqslant n(v^1_{j^1}, v^h_{j^h}) \leqslant \rho$. The case where (w.l.o.g.) $v = v^2_{b^2}$ is the largest vertex in the k-tuple goes exactly along the same lines: we omit the details. □

Corollary 2.11. *If a consistent ordering and a k-partition of $V(G)$ are given, then a maximum weighted stable set of G may be found in $O(|V(G)| \cdot (\Delta(G)+1)^{k-1})$-time.*

We close with a slight generalization of Theorems 2.5 and 2.10 which will be exploited in the application described in Section 4. We omit its proof.

Theorem 2.12. *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 $d \in \mathscr{Z}^k_+$. A maximum (minimum) weighted stable 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 $D = \max\{d^h, 1 \leqslant h \leqslant k\}$.*

3. The thinness of a graph

Definition 3.1. 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 $thin(G) \leqslant |V|$ for every graph $G(V, E)$. In fact, the partition of V into $|V|$ classes of size one (each class with one vertex) is consistent with every ordering of V. Also, the thinness of a graph G is 1 if and only if G is an interval graph. We now relate the thinness of a graph to its pathwidth (for a survey on the pathwidth and the treewidth, see [3]).

Definition 3.2. A path decomposition of a graph $G = (V, E)$ is a sequence of subsets of vertices $(X_1, X_2, \ldots, X_r)$ such that

1. $X_1 \cup \cdots \cup X_r = V$
2. for all edges $vw \in E$, there exists an i, $1 \leqslant i \leqslant r$, with $v \in X_i$ and $w \in X_i$.
3. for all $i, j, k \in I$: if $i \leqslant j \leqslant k$, then $X_i \cap X_k \subseteq X_j$.

The width of a path decomposition $(X_1, X_2, \ldots, X_r)$ is defined as $\max_i |X_i| - 1$. The pathwidth of a graph G is the minimum possible width over all possible path decompositions of G.

The pathwidth of G is bounded by $|V(G)| - 1$ and the bound is tight for cliques. It is easy to check that a graph G has a path decomposition of width q if and only if it can be obtained from an interval graph G' of clique size q+1, by deleting some edges. This is related to the following theorem (in fact, the ordering given in the proof is the interval order of $V(G')$ and the $(q+1)$-partition is an optimal coloring for G').

Theorem 3.3. *Let $G(V, E)$ be a graph together with a path decomposition $(S_1, S_2, \ldots, S_r)$ of width q. There exists a consistent ordering and a k-partition of V such that $k \leqslant q+1$ and $\rho \leqslant 1$.*

Proof. Consider an optimal path decomposition. Let $k = pathwidth(G) + 1$ be the cardinality of the biggest set. We demonstrate that the graph is k-thin.

Let $S_1, S_2, \ldots, S_t$ 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 partition consistent with the ordering.

(1) *Ordering*: For a vertex x let i be the smallest integer such that $x \in S_i$: we say that x is first-met in S_i. Our numbering is as follows. If x is first-met in S_i and y is met in S_j with $i < j$ then $x < y$. If $i = j$ then x and y can be relatively put in arbitrary order.

(2) *Assigning vertices to classes*: We keep k classes. We start with S_1 and process the vertices in it. The rule is, we assign each vertex of S_1 to a different class. Since $|S_1| \leqslant k$, this can be done. Now we take S_2 and we do the same thing. Note that some vertices in S_2 are already assigned to certain classes, but all different classes. We have only to deal with the remaining vertices of S_2, namely the vertices of $S_2 - S_1$. We assign them to classes not used by the vertices of $S_2 \cap S_1$. All of them can be assigned to different classes, since we have enough classes as can be easily seen. We carry on with this procedure.

We claim that by the process, for each i, vertices in S_i can be assigned to classes such that if $x, y \in S_i$, then x and y goes to different classes. Clearly, it is enough to show that when we reach S_i, those vertices in S_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 S_i. Let x and y be the two vertices in S_i, which happen to be in the same class at this stage. Clearly both x and y are first-met in S_j and S_k, with $j < k < i$ (without loss of generality). Then, it means x, y have together occurred before in some subset earlier than S_i, for example, in S_k. (y is in S_k by assumption; x is in S_k because it is in S_j as well as in S_i: then, by the definition of pathwidth $x \in S_k$ also.) This contradicts the fact that inconsistency was occurring for the first time in S_i.

We write down what we have achieved: if $x, y \in S_i$, for some i, then $\text{class}(x) \neq \text{class}(y)$.

Claim. *Let the vertex x be first-met in S_i. All neighbors of x lower than x are present in S_i.*

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

So, all neighbors of x which are lower than x are present in S_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 S_k such that z, y both in S_k. Thus, $y < z$ tells us that y is first-met in S_k and z is first-met in S_j with $k < j$. Also, $z < x$ tells us that $j \leqslant i$. But by the claim above $y \in S_i$, since y is a neighbor of x (and is lower than x). Thus $y \in S_k$, $y \in S_i$, but not in S_j where $k < j \leqslant i$, which is impossible by the definition of pathwidth. Thus, the partition and the ordering of $V(G)$ we obtained are consistent and $thin(G) \leqslant pathwidth(G) + 1$. □

We have two corollaries. The former is immediate (observe that the gap between thinness and pathwidth may be high since the thinness of a clique is 1). The latter uses Theorem 2.10 and the result was already known [3]. We point out that our algorithm for k-thin graphs is indeed a generalization of the weighted stable set algorithm for bounded pathwidth graphs.

Corollary 3.4. *$Thin(G) \leqslant pathwidth(G) + 1$.*

Corollary 3.5. *If a path decomposition $(X_1, X_2, \ldots, X_r)$ of width q is given for a graph G, then a maximum weighted stable set of G may be found in $O(|V(G)|2^q)$-time.*

4. An application to the frequency assignment problem

The 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 connections (more details can be found in [1,8]). In general, if u, v are two transmitters of T and f, g two frequencies in the available spectrum $F = \{1, \ldots, f_{\max}\}$, assigning f to u and g to v may cause interference: this fact is represented by

a penalty $p(u, f, v, g)$. One wants to find an assignment of frequencies to all transmitters minimizing the sum of penalty costs.

For the solution of the FAP we have designed a local search scheme. Local search proceeds by iteratively optimizing in the neighborhood of the current optimal solution until no improving solution exists [11]. Along the lines of the ideas developed in [2,4], some set of transmitters, the *hard sets*, can be exploited to define a *large-scale neighborhood search* for the FAP. Namely, for a hard set $H \subset T$ and an assignment s of T, 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 H.

As we will show, an optimal solution in $N_{|H|}(s)$ can be found very efficiently. This is very useful for our local search; on the other hand, since re-optimizing over *small portions* of large operating networks is a routine operation, this result is relevant in itself.

We now define hard sets. For special pairs of transmitters (typically those mounted on a same physical support), the penalties can be so large as to interdict the corresponding assignments. We define $u, v \in T$ as a *hard pair* if there exists $c_{uv} \geqslant 0$ such that (i) assigning $f \in F$ to u and $g \in F$ to v is infeasible for all $|f-g| < c_{uv}$ and (ii) $p(u, f, v, g)=0$ for $|f-g| \geqslant c_{uv}$. A set $H \subseteq T$ is a *hard set* if and only if u, v is a hard pair for all $u, v \in H$.

Optimizing in $N_{|H|}(s)$ is therefore equivalent to the following sub-problem. We are given a 5-tuple (H, C, d, F, w): H is the hard set of transmitters, $F = \{1, \ldots, f_{\max}\}$ is the set of available frequencies, $C \in Z_+^{|H|\times|H|}$ is the *distance requirements* matrix, $d \in Z_+^{|H|}$ is the *demand* vector (i.e. every transmitter $v \in H$ has to be assigned a number d_v of frequencies) and $w \in R_+^{|H|\times|F|}$ is the frequency costs matrix, i.e. w_{uf} denotes the (penalty) cost of assigning $f \in F$ to $u \in H$ (such cost accounts for the interference penalties with the transmitters in $T - H$).

We want to find a feasible frequency assignment of H such that the sum of frequency costs is minimized. A *feasible frequency assignment* of H 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*); and (ii) for each $u, v \in H$, if $f \in F_u$ and $g \in F_v$, $(u, g) \neq (v, f)$, then $|g - f| \geqslant c_{uv}$ (*distance constraint*).

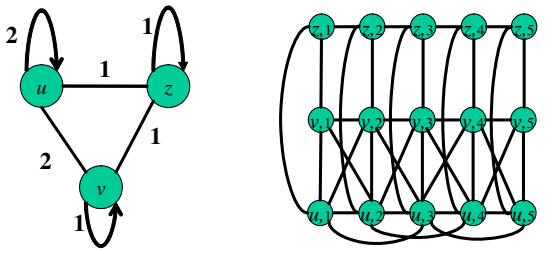

Fig. 2. An example of the conflict graph. Here, $c_{vv} = c_{zz} = c_{uz} = c_{vz} = 1$, $c_{uv} = c_{uu} = 2$.

As we show in the following, this sub-problem can be reduced to the solution of a minimum weight (cardinality constrained) stable set problem on a suitable graph $G(W, E)$ (named *conflict graph* [1]), where $W = \{(v, f) : v \in H, f \in F\}$ and $E = \{(u, g)(v, f)$: $|g - f| < c_{uv}$, (u, g) 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. An example is given in Fig. 2.

Lemma 4.1. *The graph $G(W, E)$ is $|H|$-thin.*

Proof. 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 $\mathscr{W} = \{W^{v_1}, \ldots, W^{v_n}\}$. It is immediate to verify that σ and $\mathscr{W}$ are consistent. □

Theorem 4.2. *An instance (H, C, d, F, w) of the sub-problem can be solved in $O(f_{\max} \cdot |H| \cdot (1 + \phi)^{|H|-1} \cdot (D + 1)^{|H|})$-time, where $\phi = \max_{u\in H; v,z\in H} |c_{uv} - c_{uz}|$ and $D = \max_{v\in H} d_v$.*

Proof. Let $G(W, E)$ be the conflict graph. With every subset of vertices $S \subseteq W$, we associate a frequency assignment $\mathscr{F} = \{F_1, \ldots, F_{|H|}\}$, by letting $F_v = \{f : (v, f) \in S\}$ for all $v \in H$. Moreover, if S is an stable 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 a stable 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 a stable 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.

Table 1
Computational results over COST253 instances

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

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 [5]. The times refer to a C code, running under LINUX operating system on a two processors Intel Xeon, 1.5 GHz.

So, our original task is now reduced to the problem of finding a minimum weight (cardinality constrained) stable set problem on a $|H|$-thin graph. To complete the proof we only need to verify that $\rho(\sigma, \mathscr{W}) = \phi$ and the result follows immediately from Theorem 2.12. In order to evaluate $\rho(\sigma, \mathscr{W})$, 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 than $v_{p(j)} = (v, f - c_{uv})$. 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. □

We tested our approach on the COST259 test-bed [5], a collection of large instances of the FAP in GSM

networks. The scenarios have been contributed by three industrial sources, namely E-Plus Mobilfunk GmbH, Siemens AG, and Swisscom Ltd. These problems are of high interest both from the academic and the industrial point of view. The initial solutions for the local search were found by a suitable implementation of simulated annealing; details may be found in [9]. In these instance usually $\phi \leqslant 2$, and therefore the sub-problem in Theorem 4.2 could be solved very efficiently. As detailed in Table 1, we were able to improve most of the former best known solutions (listed in column *Old best*), contributed by a number of academic and industrial research groups (see [5]). Observe that in one case (*bradford-1-race*) the improvement exceeds 60%, a very large improvement for a mobile operator. Only in two cases the algorithm did not obtain the best known solution values, however, performing only slightly worse (less than 2%) than Old Best. The large running times are motivated (for our and other approaches as well) by the huge size of real-life instances and are irrelevant in practical planning.

Acknowledgments

We are grateful to Antonio Sassano for sparking our interest in the problem. We also wish to thank Günter Rote for an insightful discussion and an anonymous referee for several helpful comments.

References

[1] K. Aardal, S.P.M. van Hoesel, A. Koster, C. Mannino, A. Sassano, Models and solution techniques for frequency assignment problems, 4OR 4 (1) (2003) 261–317.
[2] R.K. Ahuja, O. Ergun, J.B. Orlin, 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 tourist guide through treewidth, Acta Cybernet. 11 (1993) 1–21.
[4] V.G. Deineko, G.J. Woeginger, A study of exponential neighborhoods for the travelling salesman problem and for the quadratic assignment problem, Math. Program. 87 (A) (2000) 519–542.
[5] A. Eisenblätter, A. Koster, FAP web—A website about frequency assignment problems, ⟨http://fap.zib.de/⟩.
[6] U.I. Gupta, D.T. Lee, J.Y.T. Leung, Efficient algorithms for interval graphs and circular-arc graphs, Networks 12 (1982) 459–467.
[7] J.Y. Hsiao, C.Y. Tang, R. Chang, An efficient algorithm for finding a maximum weight 2-independent set on interval graphs, Information Process. Lett. 43 (5) (1992) 229–235.
[8] A. Koster, Frequency assignment—models and algorithms, Ph.D. Thesis, Maastricht University, Netherlands 1999.
[9] C. Mannino, G. Oriolo, F. Ricci, A short note on the implementation of a simulated annealing procedure for FAP, ⟨http://www.dis.uniroma1.it/~mannino/papers/SA_FAP.pdf⟩.
[10] S. Olariu, An optimal greedy heuristic to color interval graphs, Inform. Process. Lett. 37 (1991) 21–25.
[11] C. Papadimitriou, K. Steiglitz, Combinatorial Optimization, Prentice-Hall, Englewood Cliffs, NJ, 1982.
[12] J. Ramalingam, C. Pandu Rangan, A unified approach to domination problems on interval graphs, Inform. Process. Lett. 27 (1988) 271–274.