Discrete Optimization 8 (2011) 540–554

Contents lists available at SciVerse ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

A branch-and-cut algorithm for the minimum-adjacency vertex coloring problem

Diego Delle Donne, Javier Marenco *

Computer Science Department, FCEN, University of Buenos Aires, Pabellón I, Ciudad Universitaria, (1428) Buenos Aires, Argentina
Sciences Institute, National University of General Sarmiento, J.M. Gutiérrez y J. Verdi, Malvinas Argentinas, (1613) Buenos Aires, Argentina

ARTICLE INFO

Article history:
Received 25 February 2009
Received in revised form 10 May 2011
Accepted 12 May 2011
Available online 12 June 2011

Keywords:
Frequency assignment
Adjacent colors
Integer programming

ABSTRACT

In this work we study a particular way of dealing with interference in combinatorial optimization models representing wireless communication networks. In a typical wireless network, co-channel interference occurs whenever two overlapping antennas use the same frequency channel, and a less critical interference is generated whenever two overlapping antennas use adjacent channels. This motivates the formulation of the *minimum-adjacency vertex coloring problem* which, given an interference graph G representing the potential interference between the antennas and a set of prespecified colors/channels, asks for a vertex coloring of G minimizing the number of edges receiving adjacent colors. We propose an integer programming model for this problem and present three families of facet-inducing valid inequalities. Based on these results, we implement a branch-and-cut algorithm for this problem, and we provide promising computational results.

1. Introduction

In this work, we are interested in a combinatorial optimization problem arising from frequency assignment problems in wireless communication networks, that was motivated by the types of interference generated in GSM mobile phone networks [1].

A wireless network employs some portion of the electromagnetic spectrum to establish communications between the transmitter/receiver network antennas, called TRXs. A certain part of the electromagnetic spectrum is licensed to the company operating the network and is divided into discrete channels. Each TRX must operate through one channel, although whenever two TRXs overlap their coverage areas *co-channel interference* occurs if both are using the same channel, and communications cannot be established within the common area. Moreover, if these conflicting TRXs are assigned to adjacent channels, then the so-called *adjacent-channel interference* occurs, generating in this case a minor interference only. In a typical scenario, a good channel assignment *must* avoid co-channel interference and *should* avoid adjacent-channel interference.

Several other constraints arise in practical settings as, e.g., blocked channels and separation constraints (see, e.g., [1–4]) but in this work we focus on the basic model as stated in the previous paragraph. We are interested in the polyhedral structure generated by such a combinatorial optimization problem, which includes a graph coloring structure with additional considerations on adjacent channels/colors. Based on these observations, we introduce in this work the *minimum-adjacency vertex coloring problem*, present an initial polyhedral study and, based on these results, implement a branch-and-cut algorithm for this problem.

* Corresponding author at: Computer Science Department, FCEN, University of Buenos Aires, Pabellón I, Ciudad Universitaria, (1428) Buenos Aires, Argentina. Tel.: +54 11 4863 8920.
E-mail addresses: ddelle@dc.uba.ar (D. Delle Donne), jmarenco@dc.uba.ar (J. Marenco).

1572-5286/$ – see front matter
doi:10.1016/j.disopt.2011.05.003

This paper is organized as follows. In Section 2 we formally state the minimum-adjacency vertex coloring problem and provide an integer programming model for this problem. Section 3 introduces three families of facet-inducing valid inequalities for this problem. Section 4 describes the implementation details of our branch-and-cut algorithm, and Section 5 provides some computational experiments. The paper closes in Section 6 with some concluding remarks and future work.

2. Problem formulation and integer programming model

We first introduce the *interference graph* $G = (V, E)$ associated with an instance, in such a way that V represents the set of TRXs in the network and, whenever the coverage areas of two TRXs overlap, an edge in E joins the corresponding vertices. Throughout this work we shall use the notation $n = |V|$ and $m = |E|$. Let $C = \{1, \dots, t\}$ be a set of consecutive colors representing the t available channels. A *C-coloring* of G is a function $c : V \to C$ such that $c(i) \neq c(j)$ for every $ij \in E$. Clearly, a C-coloring of G corresponds to a feasible frequency assignment avoiding co-channel interference. Finally, for every $vw \in E$, we define $\psi(vw) \in [0, 1]$ to be the level of interference generated when the vertices v and w are assigned adjacent colors/channels.

Minimum-adjacency vertex coloring problem. *Given an interference graph $G = (V, E)$, a set of consecutive colors $C = \{1, \dots, t\}$, and an interference function $\psi : E \to [0, 1]$, find a C-coloring of G minimizing the total adjacent-channel interference, i.e.,*

$$\min_{y \in \mathcal{C}(G,C)} \sum \{\psi(vw) : vw \in E \text{ and } |y(v) - y(w)| = 1\},$$

where $\mathcal{C}(G, C)$ represents the set of all C-colorings of G.

In the minimum-adjacency vertex coloring problem we ask for a frequency assignment with no co-channel interference (i.e., a C-coloring of G) minimizing the less critical adjacent-channel interference. This provides a good compromise between forbidding both kinds of interference and allowing them with a penalty term in the objective function. This problem is clearly $\mathcal{NP}$-hard since it generalizes the classical vertex coloring problem, thus motivating the integer programming approach started in this work.

In real-world instances, the number of available frequencies may not be enough to cover the whole network with a co-channel free assignment. This issue can be addressed by allowing co-channel interference between some prespecified pairs of adjacent vertices, while penalizing such interference in the objective function. For the sake of simplicity we do not further develop this idea in the current work. When all co-channel interference is allowed, we obtain the *minimum interference frequency assignment problem*. We refer to [5] for a complete survey of further variants of frequency assignment problems. A relaxation of this type of problems using semidefinite programming is studied by Eisenblätter in [6], obtaining very strong lower bounds.

Throughout this work we shall consider $\psi(vw) = 1$ for every $vw \in E$, so the minimum-adjacency vertex coloring problem reduces to finding a C-coloring of G which minimizes the number of edges receiving adjacent colors. Note that the adjacencies within the color set C are not circular, i.e., the colors 1 and t are not adjacent if $t \geq 3$.

In order to state an integer programming formulation for the minimum-adjacency vertex coloring problem, for every $v \in V$ and every $c \in C$ we introduce the binary *assignment variable* x_{vc} representing whether the color c is assigned to the vertex v or not. We also introduce, for every $vw \in E$, $v < w$, the binary *adjacency variable* z_{vw} asserting whether the vertices v and w receive adjacent colors or not. With these definitions, a model for the minimum-adjacency vertex coloring problem is given by:

$$\min \sum_{vw \in E} \psi(vw)\, z_{vw}$$

$$\sum_{c \in C} x_{vc} = 1 \quad \forall v \in V \tag{1}$$

$$x_{vc} + x_{wc} \leq 1 \quad \forall vw \in E,\ v < w,\ \forall c \in C \tag{2}$$

$$x_{vc_1} + x_{wc_2} \leq 1 + z_{vw} \quad \forall vw \in E,\ v < w,\ \forall c_1, c_2 \in C,\ |c_1 - c_2| = 1 \tag{3}$$

$$x_{vc} \in \{0, 1\} \quad \forall v \in V,\ \forall c \in C \tag{4}$$

$$z_{vw} \in \{0, 1\} \quad \forall vw \in E,\ v < w. \tag{5}$$

Constraints (1) ensure that every vertex is assigned exactly one color from C and constraints (2) forbid adjacent vertices to be assigned the same color. Constraints (3) force adjacent vertices to be assigned nonadjacent colors unless the corresponding adjacency variable takes the value 1. Finally, constraints (4) and (5) force the variables to be binary. We call this formulation the *stable-set model*, which is a straightforward adaptation of the formulation presented in [7,8] for the classical vertex coloring problem.

Note that the constraints allow feasible solutions to have active adjacency variables even when the corresponding vertices are not assigned to adjacent colors. We say that $y = (x, z)$ is an *undominated* solution if $z_{vw} = 0$ for every $vw \in E$ such that v and w are assigned to nonadjacent colors. Note that any optimal solution is undominated if $\psi(vw) > 0$ for every $vw \in E$.

2.1. Alternative formulations

The *orientation model* for the classical vertex coloring problem [9] can also be adapted to the minimum-adjacency vertex coloring problem in a straightforward way. In this model, for every vertex $v \in V$ we employ an integer variable $x_v \in \{1, \ldots, t\}$ containing the color assigned to v (recall that $t = |C|$). We also have for every $vw \in E$, $v < w$, binary *orientation variables* o_{vw} and o_{wv} in such a way that $o_{vw} = 1$ if and only if $x_v < x_w$ and, finally, we employ the adjacency variables as in the stable-set model.

We have also considered a new formulation for vertex coloring problems, namely the *distance model*, consisting of *distance variables* $x_{vw} \in [-t+1, t-1]$ containing the distance between the colors assigned to v and w, for every $v, w \in V$, $v < w$. We also employ in this model the orientation variables and the adjacency variables. This model is, to the best of our knowledge, a new approach for vertex coloring problems, which can easily be adapted to the classical vertex coloring problem.

We have performed computational experiments in order to compare the performance of these integer programming formulations with a pure branch-and-bound procedure. In these experiments we resorted to the instances introduced in Section 5. According to the results, the proposed stable-set model outperforms both the orientation model and the distance model, hence suggesting that the stable-set model is a good starting point for the implementation of a branch-and-cut algorithm. In Section 3 we start such an approach by studying the polytope associated with this integer programming model.

To conclude this section, it is interesting to note that the *representatives model* introduced in [10] cannot be adapted to the minimum-adjacency vertex coloring problem in a straightforward way, since it admits no distance notion among the assigned colors. This notion can be incorporated to the model by explicitly identifying the color assigned to each vertex but, unfortunately, this addition greatly affects the model structure. It is also interesting to mention that the classical column generation approach for the vertex coloring problem presented in [11] can be adapted to a frequency assignment setting in order to take into account the color adjacencies generated by the color assignment, as was performed in [12,13].

3. Polyhedral study

In this section we introduce the polytope associated with the stable-set formulation. We characterize the dimension of this polytope for $|C| > \chi(G)$, and we present three families of facet-inducing valid inequalities.

Definition 1. Given a graph $G = (V, E)$ and a set of colors C, we define $PS(G, C)$ to be the convex hull of the incidence vectors $(x, z) \in \mathbb{R}^{nt+m}$ of feasible solutions to the stable-set model (1)–(5).

Proposition 1. *If $|C| > \chi(G)$ and $E \neq \emptyset$, then* $\dim(PS(G, C)) = n(t-1) + m$*, and a minimal equation system is defined by* (1).

Proof. Let $\lambda \in \mathbb{R}^{nt+m}$ and $\lambda_0 \in \mathbb{R}$ such that $\lambda y = \lambda_0$ for every feasible solution $y \in PS(G, C)$. It suffices to show that (λ, λ_0) is a linear combination of the model constraints (1).

Let $vw \in E$, and consider a feasible solution $y = (x, z) \in PS(G, C)$ such that the colors assigned to v and w are nonadjacent and, furthermore, $z_{vw} = 0$. Such a solution can always be constructed since $|C| > \chi(G) \geq 2$ (as $E \neq \emptyset$). Let $y' = (x, z') \in PS(G, C)$ be the feasible solution obtained from y by setting $z'_{vw} = 1$ and leaving the remaining variables unchanged. The solution y' is clearly feasible and only differs from y in the z_{vw}-variable, hence $\lambda_{z_{vw}} = 0$.

Let $v \in V$. Since $|C| > \chi(G)$, there exists a feasible solution $y = (x, z) \in PS(G, C)$ such that at least one color from C is not used by any vertex. Let $c \in C$ be the color assigned to the vertex v by y, and let $c' \in C$ be a color not used in y. Define $y' = (x', z') \in PS(G, C)$ to be the feasible solution obtained from y by setting $x'_{vc} = 0$, $x'_{vc'} = 1$, and leaving the remaining x-variables unchanged. The solution y' is feasible as the color c' is not used in y and, furthermore, x only differs from x' in the x'_{vc}-and $x'_{vc'}$-variables, and possibly some z-variables. Since $\lambda_z = 0$, we conclude $\lambda_{x_{vc}} = \lambda_{x_{vc'}}$. By arbitrarily renaming the colors, we conclude that $\lambda_{x_{vc}} = \lambda_{x_{vc'}}$ for any $c, c' \in C$.

By combining these observations, we conclude that λ is a linear combination of the coefficient vectors of the model constraints (1). Since the coefficient vectors of these constraints are linearly independent, we conclude that (1) is a minimal equation system for $PS(G, C)$, so $\dim(PS(G, C)) = n(t-1) + m$. □

When $|C| = \chi(G)$, the dimension of the polytope strongly depends on the graph structure. Unfortunately, we do not have a complete characterization of $\dim(PS(G, C))$ in this case and, to the best of our knowledge, the same holds for all known integer programming formulations of graph coloring which explicitly include a prespecified set of colors. Due to this fact, we are not able in this work to establish facetness results when $|C| = \chi(G)$.

If $|C| > \chi(G)$, it can be easily proved that the only facet-inducing constraints from the model are the relaxed constraints $0 \leq z_{vw}$ and $z_{vw} \leq 1$ for every $vw \in E$, and the non-negativity constraints for the assignment variables.

3.1. Consecutive colors clique inequalities

We introduce in this section a family of valid inequalities dominating the model constraints (3), which showed to be particularly useful in the branch-and-cut algorithm. These inequalities measure the number of adjacencies generated when some vertices from a clique in G are assigned into a set of consecutive colors. In this work we consider a *clique* to be a (not necessarily maximal) set of pairwise adjacent vertices. If $v \in V$, we define $\mathcal{N}(v) = \{w \in V : vw \in E\}$ to be the set of neighbors of the vertex v and, if $A \subseteq V$, we define $\mathcal{N}_A(v) = \mathcal{N}(v) \cap A$ to be the set of neighbors in A of the vertex v.

Definition 2 (*Consecutive Colors Clique Inequalities*). Let $K \subseteq V$ be a clique of G, and let $Q = \{c_1, \ldots, c_q\} \subseteq C$ be a set of consecutive colors such that $c_{i+1} = c_i + 1$, for $i = 1, \ldots, q-1$. We define the *consecutive colors clique inequality* associated with K and Q to be

$$\sum_{i \in K} \left[x_{ic_1} + x_{ic_q} + \sum_{c \in Q \setminus \{c_1, c_q\}} 2x_{ic} \right] \le (q-1) + \sum_{i,j \in K} z_{ij}. \tag{6}$$

Note that for every $vw \in E$, if we define $K = \{v, w\}$ and we use only two adjacent colors, i.e., $|Q| = \{c_1, c_2\}$, then the resulting consecutive colors clique inequality dominates the constraints (3) corresponding to the edge vw, and the colors c_1 and c_2.

In order to prove validity of (6), we need the following notation. Given a C-coloring y of G, if $Q \subseteq C$ is a set of consecutive colors and $W \subseteq V$ is a set of vertices, we say that $Q_1^y, \ldots, Q_r^y \subseteq Q$ is a *color distribution of* Q *in* W *given by* y, if $Q_i^y \subseteq Q$ is a set of consecutive colors for $i = 1, \ldots, r$, and $c \in Q_i^y$ if and only if $x_{vc} = 1$ for some $v \in W$. In other words, a color distribution of Q in W is a partition of the colors from Q assigned to W into sets of consecutive colors. We say that the color distribution $Q_1^y, \ldots, Q_r^y$ is *minimal* if Q_i^y and Q_j^y do not admit consecutive colors, for $i, j \in \{1, \ldots, r\}$, $i \ne j$. It is not difficult to verify that the minimal color distribution of Q in W is unique up to set permutations.

We say that $\overline{Q}_1^y, \ldots, \overline{Q}_{\overline{r}}^y$ is an *inverse color distribution of* Q *in* W *given by* y, if $\overline{Q}_i^y \subseteq Q$ is a set of consecutive colors for $i = 1, \ldots, \overline{r}$, and $c \in \overline{Q}_i^y$ if and only if $x_{vc} = 0$ for each $v \in W$. In other words, an inverse color distribution of Q in W is a partition of the colors from Q not assigned to W into sets of consecutive colors. We say that the inverse color distribution $\overline{Q}_1^y, \ldots, \overline{Q}_{\overline{r}}^y$ is *minimal* if $\overline{Q}_i^y$ and $\overline{Q}_j^y$ do not admit consecutive colors, for $i, j \in \{1, \ldots, \overline{r}\}$, $i \ne j$.

If $y = (x, z) \in PS(G, C)$ is a feasible solution, we say that the variable x_{vc} is *active* in y if $x_{vc} = 1$, for $v \in V$ and $c \in C$. Similarly, we define the variable z_{vw} to be *active* in y if $z_{vw} = 1$, for $vw \in E$.

Proposition 2. *The consecutive colors clique inequalities* (6) *are valid for* $PS(G, C)$.

Proof. Let $y = (x, z) \in PS(G, C)$ be an integer feasible solution and, for $j = 1, \ldots, q$, define $\delta_j = 1$ if the color c_j is assigned to some vertex of K and $\delta_j = 0$ otherwise. Let $Q_1^y, \ldots, Q_r^y$ be a minimal color distribution of Q in K given by y. We can state the LHS of (6) as

$$\sum_{v \in K} \left[x_{vc_1} + x_{vc_q} + \sum_{c \in Q \setminus \{c_1, c_q\}} 2x_{vc} \right] = \left[\sum_{i=1}^{r} 2|Q_i^y| \right] - \delta_1 - \delta_q, \tag{7}$$

since the only active variables among x_{vc}, for $v \in K$ and $c \in Q$, are some variables corresponding to colors in $Q_1^y, \ldots, Q_r^y$, and each of these colors is assigned to exactly one vertex in K, since K is a clique. We subtract δ_1 resp. δ_q in the RHS of (7) so the color c_1 resp. c_q contributes with one unit (instead of two) to the LHS of (6). On the other hand, we can bound the RHS of (6) by

$$(q-1) + \sum_{v,w \in K} z_{vw} \ge (q-1) + \sum_{i=1}^{r} (|Q_i^y| - 1) = (q-1) - r + \sum_{i=1}^{r} |Q_i^y|, \tag{8}$$

since Q_i^y generates exactly $|Q_i^y| - 1$ active adjacency variables, for $i = 1, \ldots, r$.

The number of colors from Q assigned to vertices in K is $\sum_{i=1}^{r} |Q_i^y|$. Moreover, since the color distribution $Q_1^y, \ldots, Q_r^y$ is minimal, then no pair of sets Q_i^y and Q_j^y admit consecutive colors, so there exist at least $r-1$ colors in $Q \setminus \{c_1, c_q\}$ not assigned to vertices in K. Combining these observations, we conclude

$$\sum_{i=1}^{r} |Q_i^y| + (r-1) + (1-\delta_1) + (1-\delta_q) \le q, \tag{9}$$

as $(1-\delta_j) = 1$ only when the color c_j is not assigned to any vertex from K. By adding $\sum_{i=1}^{r} |Q_i^y|$ to each side of (9) we obtain

$$\left[\sum_{i=1}^{r} 2|Q_i^y| \right] - \delta_1 - \delta_q \le (q-1) - r + \sum_{i=1}^{r} |Q_i^y|.$$

By combining (7), (8), and (9), we conclude that (6) holds for y. Since y is an arbitrary feasible solution, the inequality (6) is valid for $PS(G, C)$. $\square$

We now show that the consecutive colors clique inequalities induce facets of $PS(G, C)$ under suitable hypotheses. We first state a preliminary lemma characterizing the feasible solutions satisfying these inequalities with equality.

Lemma 1. *If $|C| > 2|K| - |Q| + 1$, then a feasible solution $y \in PS(G, C)$ satisfies (6) with equality if and only if*

(i) *for each pair $\{c_i, c_{i+1}\}$ of consecutive colors in Q, at least one of them is assigned to some vertex in K,*
(ii) *there are no adjacency variables $z_{vw} = 1$, with $v, w \in K$, such that v or w are assigned colors in $C \setminus Q$, and*
(iii) *there are no adjacency variables $z_{vw} = 1$, with $v, w \in K$, such that v and w are assigned nonadjacent colors.*

Proof. Assume first that the integer feasible solution $y = (x, z) \in PS(G, C)$ satisfies (6) with equality. If the condition (iii) does not hold, then there exists some active-adjacency variable $z_{vw} = 1$, with $v, w \in K$, such that v or w are assigned nonadjacent colors. If we set $z_{vw} = 0$, the resulting solution is still feasible but the RHS of (6) is smaller, hence y does not satisfy (6) with equality, a contradiction.

If the condition (iii) holds but the condition (ii) does not hold, then there exists some active-adjacency variable $z_{vw} = 1$, with $v, w \in K$, such that v or w are assigned colors outside Q. Assume w.l.o.g. that $x_{vc} = 1$, for $c \in C \setminus Q$. If all colors in Q are used by vertices from K, reorder the colors in $C \setminus Q$ in such a way that no two vertices from K receiving colors outside Q are assigned adjacent colors. Such a reordering is possible if there are at least $2(|K| - |Q|) + 1$ colors in $C \setminus Q$, which is guaranteed by the hypothesis. After this operation we can set $z_{vw} = 0$, hence decreasing the RHS of (6), which implies that y does not satisfy (6) with equality, a contradiction.

Now, if there exists some color $c' \in Q$ not assigned to any vertex in K, it is possible to swap the color classes corresponding to the colors c and c'. Suppose $c' = c_1$ and assume w.l.o.g. that w is not assigned to the color preceding c_1 (if it is we can swap the color classes corresponding to v and w). If we set $x_{vc_1} = 1$ and $z_{vw} = 0$, we increase the LHS of (6) by one unit. Furthermore, the RHS is not increased even if $x_{w'c_2} = 1$ for some $w' \in K$, since z_{vw} is now equal to 0. The same happens if $c' = c_q$ assuming that w is not assigned to the color following c_q. Suppose now that $c' \in Q \setminus \{c_1, c_q\}$, then setting $x_{vc'} = 1$ and $z_{vw} = 0$ will increase the LHS of (6) by 2. Furthermore, the RHS increases by at most one unit even if both adjacent colors to c' are assigned to vertices in K, since z_{vw} is again now equal to 0. After this operation, the difference between the RHS and the LHS of (6) is reduced (and the obtained solution is feasible), hence y does not satisfy (6) with equality, a contradiction.

Therefore, if the conditions (ii) or (iii) do not hold then y does not satisfy (6) with equality, so assume that (ii) and (iii) are met. Let $Q_1^y, \dots, Q_r^y$ be a minimal color distribution of Q in K, and let $\overline{Q}_1^y, \dots, \overline{Q}_{\overline{r}}^y$ be a minimal inverse color distribution of Q in K. Since the only active x-variables in (6) are the variables corresponding to colors in $Q_1^y, \dots, Q_r^y$, and each such color is assigned to exactly one vertex in K, we can rewrite the LHS of (6) as

$$\sum_{v \in K} \left[x_{vc_1} + x_{vc_q} + \sum_{c \in Q \setminus \{c_1, c_q\}} 2x_{vc} \right] = \left[\sum_{i=1}^{r} 2|Q_i^y| \right] - \delta_1 - \delta_q, \tag{10}$$

where $\delta_j = 1$ if the color c_j is assigned to some vertex in K and $\delta_j = 0$ otherwise. Furthermore, the color distribution $Q_1^y, \dots, Q_r^y$ and the inverse color distribution $\overline{Q}_1^y, \dots, \overline{Q}_{\overline{r}}^y$ define a partition of Q, thus

$$|Q| = q = \sum_{i=1}^{r} |Q_i^y| + \sum_{j=1}^{\overline{r}} |\overline{Q}_j^y|. \tag{11}$$

Finally, by (ii) and (iii) we can assume that $z_{vw} = 1$ for $v, w \in K$ only if v and w are assigned colors from Q, therefore

$$\sum_{v,w \in K} z_{vw} = \sum_{i=1}^{r} (|Q_i^y| - 1). \tag{12}$$

By replacing (10), (11), and (12) in the expression of the valid inequality (6), we conclude that (6) is satisfied with equality if and only if

$$\left[\sum_{i=1}^{r} 2|Q_i^y| \right] - \delta_1 - \delta_q = \left(\sum_{i=1}^{r} |Q_i^y| + \sum_{j=1}^{\overline{r}} |\overline{Q}_j^y| \right) - 1 + \sum_{i=1}^{r} (|Q_i^y| - 1). \tag{13}$$

Restating this equation, we obtain

$$\sum_{j=1}^{\overline{r}} |\overline{Q}_j^y| = r + 1 - \delta_1 - \delta_q = (r - 1) + (1 - \delta_1) + (1 - \delta_q). \tag{14}$$

Therefore, the number of colors from $Q \setminus \{c_1, c_q\}$ not assigned to vertices in K is exactly $r - 1$, and this holds if and only if $\overline{Q}_i^y$ is a singleton, for $i = 1, \dots, \overline{r}$ or, equivalently, at least one color from each pair of consecutive colors in Q is assigned to some vertex in K. □

Theorem 1. *If $|C| > \chi(G)$, $|C| > |Q|$, $|C| \geq |K| + 5$, $|C| \geq 2|K| - |Q| + 3$, and $|K| \geq \frac{|Q|}{2} + 1$, then the consecutive colors clique inequality (6) induces a facet of $PS(G, C)$.*

		c		c_1	c_2	$\dots$	c_q	
$y \rightarrow$	$\emptyset_K$	$\emptyset$	$\emptyset_K$ $\qquad$ $\emptyset_K$	v	w			
$y' \rightarrow$	$\emptyset_K$	v	$\emptyset_K$ $\qquad$ $\emptyset_K$		w			

(a) Pair of solutions used in Claim 3.

		c'		c_1	$\dots$		c		$\dots$	c_q	
$y \rightarrow$	$\emptyset_K$	$\emptyset$	$\emptyset_K$			w_1	v	w_2			
$y' \rightarrow$	$\emptyset_K$	v	$\emptyset_K$			w_1		w_2			

(b) Pair of solutions used in Claim 4.

Fig. 1. Constructions for the proof of Theorem 1.

Proof. Let $(\pi, \pi_0) \in \mathbb{R}^{nt+m+1}$ be the coefficient vector of (6), and let F be the face of $PS(G, C)$ defined by (6). Let $\lambda \in \mathbb{R}^{nt+m}$ and $\lambda_0 \in \mathbb{R}$ such that $\lambda y = \lambda_0$ for every $y \in F$. We shall verify that λ is a linear combination of π and the coefficient vectors of the model constraints (1), hence proving that F is a facet of $PS(G, C)$.

Claim 1 ($\lambda_{z_{vw}} = 0$ *for every* $vw \in E(V \setminus K)$)**.** *Let $y = (x, z) \in PS(G, C)$ be a feasible solution satisfying (6) with equality such that v and w are not assigned adjacent vertices (such a solution exists as $|C| > \chi(G)$), and construct the solution $y' = (x, z')$ only differing from y in $z'_{vw} = 1$. This new solution is feasible and satisfies (6) with equality, as z_{vw} does not appear in (6). Therefore, $\lambda y = \lambda_0 = \lambda y'$, hence $\lambda_{z_{vw}} = 0$.* $\square$

Claim 2 ($\lambda_{x_{vc}} = \lambda_{x_{vc'}}$ *for every* $v \notin K$ *and every* $c, c' \in C$)**.** *Since $|C| > \chi(G)$ we can construct a feasible solution $y = (x, z) \in PS(G, C)$ satisfying (6) with equality and such that the color c' is not assigned to any vertex, and such that $x_{vc} = 1$. Let $y' = (x', z')$ be the feasible solution obtained from y by setting $x'_{vc'} = 1$ and $x'_{vc} = 0$, and modifying the variables z'_{vw}, for $w \in \mathcal{N}(v)$ accordingly. Note that y' is feasible as c' is not assigned to any vertex in y. Since both y and y' satisfy (6) with equality and Claim 1 implies $\lambda_{z_{vw}} = 0$ for $w \in \mathcal{N}(v)$, the claim follows.* $\square$

Claim 3 ($\lambda_{x_{vc_1}} = \lambda_{x_{vc_q}} = \lambda_{x_{vc}} - \lambda_{z_{vw}}$ *for every* $v, w \in K$ *and* $c \notin Q$)**.** *Let $y = (x, z) \in PS(G, C)$ be a feasible solution satisfying (6) with equality and such that $x_{vc_1} = 1$ and $x_{wc_2} = 1$ for some $w \in K$. Further, we assume that there exists some color $c \notin Q$ not used by y, as $|C| > \chi(G)$, and that the colors adjacent to c are not assigned to any vertex in K. We need $|C| \geq 2|K| - |Q| + 3$ for this construction, which is guaranteed by the hypotheses. Let $y' = (x', w')$ be the feasible solution obtained from y by setting $x'_{vc} = 1$ and adjusting the adjacency variables accordingly (see Fig. 1 (a) for an illustration of these solutions, where the symbol $\emptyset_K$ denotes that no vertex from K is assigned the corresponding color). The solution y' is feasible by construction and, moreover, by Lemma 1 it also satisfies (6) with equality. The solutions y and y' only differ in the variables x_{vc_1}, x_{vc}, z_{vw}, and possibly some adjacency variables with vertices in $\mathcal{N}_{V\setminus K}(v)$. Claim 1 implies $\lambda_{z_{vu}} = 0$ for every $u \in \mathcal{N}_{V\setminus K}(v)$, hence $\lambda_{x_{vc_1}} + \lambda_{z_{vw}} = \lambda_{x_{vc}}$. By repeating the procedure with the colors c_q and c_{q-1} instead of c_1 and c_2, respectively, we obtain $\lambda_{x_{vc_q}} + \lambda_{z_{vw}} = \lambda_{x_{vc}}$. By arbitrarily renaming the colors and arbitrarily selecting w, we conclude that $\lambda_{x_{vc_1}} = \lambda_{x_{vc_q}} = \lambda_{x_{vc}} - \lambda_{z_{vw}}$ for every $v, w \in K$ and every $c \notin Q$.* $\square$

Note that Claim 3 implies $\lambda_{x_{vc}} = \lambda_{x_{vc'}}$ for $v \in K$ and $c, c' \notin Q$, and $\lambda_{z_{vw_1}} = \lambda_{z_{vw_2}}$ for all $v, w_1, w_2 \in K$.

Claim 4 ($\lambda_{x_{vc}} = \lambda_{x_{vc'}} - \lambda_{z_{vw_1}} - \lambda_{z_{vw_2}}$ *for every* $v \in K, c \in Q \setminus \{c_1, c_q\}, c' \notin Q$, *and every* $w_1, w_2 \in K$)**.** *Let $y = (x, z) \in PS(G, C)$ be a feasible solution satisfying (6) with equality and such that $x_{vc} = 1$. We also assume that w_1 and w_2 are assigned the two colors adjacent to c. Further, we assume that there exists some color $c' \notin Q$ not assigned to any vertex, and that the colors adjacent to c' are not assigned to any vertex from K (see Fig. 1 (b)). This assumption is feasible as $|C| \geq 2|K| - |Q| + 3$ by hypothesis. Let $y' = (x', z')$ be the feasible solution obtained from y by setting $x'_{vc'} = 1$. Again, the solution y' is valid since the color c is not assigned to any vertex in y, and Lemma 1 implies that y' satisfies (6) with equality. The solutions y and y' only differ in the variables $x_{vc}, x_{vc'}, z_{vw_1}, z_{vw_2}$, and possibly some additional adjacency variables z_{vu} with $u \notin K$ for which $\lambda_{z_{vu}} = 0$. Hence Claim 1 implies $\lambda_{x_{vc}} = \lambda_{x_{vc'}} - \lambda_{z_{vw_1}} - \lambda_{z_{vw_2}}$. By arbitrarily renaming the colors and arbitrarily selecting w_1 and w_2, we conclude that $\lambda_{x_{vc}} = \lambda_{x_{vc'}} - \lambda_{z_{vw_1}} - \lambda_{z_{vw_2}}$ for every $v \in K, c \in Q \setminus \{c_1, c_q\}, c' \notin Q$, and every $w_1, w_2 \in K$.* $\square$

Note that Claim 4 implies $\lambda_{x_{vc}} = \lambda_{x_{vc''}}$ for all $c, c'' \in Q \setminus \{c_1, c_q\}$.

Now, for every $v \notin K$, we define $\beta_v = \lambda_{x_{vc}}$, where $c \in C$ is an arbitrary color. Note that the definition of β_v does not depend on the particular choice of c, by Claim 2. On the other hand, for every $v \in K$, define $\beta_v = \lambda_{x_{vc}}$, where $c \in C \setminus Q$. Again, the definition of β_v does not depend on the choice of c, since Claim 3 implies $\lambda_{x_{vc}} = \lambda_{x_{vc'}}$ for every $v \in K$ and every $c, c' \notin Q$. Finally, define $\alpha = \lambda_{z_{vw}}$ for some $v, w \in K$, which is well defined as Claim 3 also implies $\lambda_{z_{v_1w_1}} = \lambda_{z_{v_2w_2}}$ for every $v_1, v_2, w_1, w_2 \in K$.

Claim 3 implies $\lambda_{x_{vc_1}} = \lambda_{x_{vc_q}} = \beta_v - \alpha$ for every $v \in K$ which, combined with Claim 4 yields $\lambda_{x_{vc}} = \beta_v - 2\alpha$ for each $v \in K$ and $c \in Q \setminus \{c_1, c_q\}$. These observations imply

$$\lambda = \sum_{v \in V} \beta_v \eta^{(v)} - \alpha\pi,$$

where $\eta^{(v)}$ represents the coefficient vector of the model constraint (1), for $v \in V$. Therefore, the consecutive colors clique inequality (6) induces a facet of $PS(G, C)$. □

Next we present a large family of valid inequalities generalizing (6).

Definition 3. Let $K \subseteq V$ be a clique of G. Let $Q^1 = \{c_1^1, \ldots, c_{q_1}^1\}, \ldots, Q^p = \{c_1^p, \ldots, c_{q_p}^p\} \subseteq C$ be p disjoint sets of consecutive colors such that for every $h \in \{1, \ldots, p\}$, $c_{i+1}^h = c_i^h + 1$ for $i = 1, \ldots, q_h - 1$. We define the *multi-consecutive colors clique inequality* associated with K and $Q^1, \ldots, Q^p$ to be

$$\sum_{h=1}^{p} \sum_{i \in K} \left[x_{ic_1^h} + x_{ic_{q_h}^h} + \sum_{c \in Q^h \setminus \{c_1^h, c_{q_h}^h\}} 2x_{ic} \right] \leq \sum_{h=1}^{p} (q_h - 1) + \sum_{i,j \in K} z_{ij}. \tag{15}$$

Note that Q^h and Q^{h+1} may be adjacent sets but in that case the resulting inequality is dominated by the multi-consecutive colors clique obtained by using $Q^h \cup Q^{h+1}$ as a unique set. The proof of validity of these inequalities is very similar to the proof of Proposition 2, and is therefore omitted.

Proposition 3 (*[14]*)**.** *The multi-consecutive colors clique inequalities (15) are valid for $PS(G, C)$.*

Note that the multi-consecutive colors clique inequalities are not dominated by the sum of the consecutive colors clique inequalities associated to $Q^1, \ldots, Q^p$ since the sum of the adjacency variables over the edges in K appears only once in (15). Moreover, computational results obtained using these inequalities outperformed the ones using (6). We conjecture that these inequalities define facets of $PS(G, C)$, under similar hypothesis as in Theorem 1.

3.2. Consecutive colors inner clique inequalities

In this and the following section we introduce two classes of facet-inducing inequalities for $PS(G, C)$ arising from a clique $K \subseteq V$ and a distinguished vertex $k \in K$. In these constructions, the clique structure is crucial both for validity and facetness, suggesting the importance of the cliques of the interference graph for the structure of $PS(G, C)$. The proof of facetness of the inequalities presented in this and the following section go along the same lines as the proof of Theorem 1, and are therefore omitted.

Let $K \subseteq V$ be a clique, $k \in K$ be some vertex of K, and $\{c_1, c_2, c_3\} \subseteq C$ be a set of three consecutive colors. If some vertex $v \in K \setminus \{k\}$ receives the color c_2 then an adjacency with k is generated if the latter receives either c_1 or c_3. This idea can be generalized for a set $Q = \{c_1, \ldots, c_q\} \subseteq C$ of an odd number of consecutive colors; if vertices from $K \setminus \{k\}$ are using every color $c_i \in Q$ with an odd index i, then any color $c_j \in Q$ with an even index j generates adjacencies when assigned to k. The following family of valid inequalities arises from this observation.

Definition 4 (*Consecutive Colors Inner Clique Inequality*)**.** Let $K \subseteq V$ be a clique of G, and fix a vertex $k \in K$. Let $Q = \{c_1, \ldots, c_q\} \subseteq C$, with an odd q, be a set of consecutive colors such that $c_{i+1} = c_i + 1$ for $i = 1, \ldots, q - 1$. Let $I = \{c_1, c_3, \ldots, c_q\}$ and $P = \{c_2, c_4, \ldots, c_{q-1}\}$. We define the *consecutive colors inner clique inequality* associated with K, k and Q to be

$$\left(x_{kc_1} + x_{kc_q} + \sum_{c_i \in I \setminus \{c_1, c_q\}} 2x_{kc_i} + \sum_{c_j \in P} x_{kc_j} \right) + \sum_{v \in K \setminus \{k\}} \sum_{c_j \in P} x_{vc_j} \leq \frac{q-1}{2} + \sum_{v \in K \setminus \{k\}} z_{vk}. \tag{16}$$

Note that if $Q = \{c_1, c_2, c_3\}$, $K = \{v, w\}$ and $k = v$ then the resulting consecutive colors inner clique inequality dominates the constraint (3) corresponding to the edge vw and the colors c_1 and c_2 (resp. the edge vw and the colors c_3 and c_2).

Proposition 4. *The consecutive colors inner clique inequalities (16) are valid for $PS(G, C)$.*

Proof. Let $y = (x, z) \in PS(G, C)$ be a feasible solution. For $c \in C$, define $\delta_c \in \{0, 1\}$ to be

$$\delta_c = \sum_{v \in K \setminus \{k\}} x_{vc},$$

i.e., $\delta_c = 1$ if and only if c is assigned to a vertex from $K \setminus \{k\}$. For any color $c_j \in P$, if $\delta_{c_j}(x_{kc_{j-1}} + x_{kc_{j+1}}) = 1$, then a color adjacency exists between k and some vertex from $K \setminus \{k\}$. Since K is a clique, we can bound the RHS of (16) by

$$\frac{q-1}{2} + \sum_{c_j \in P} \delta_{c_j}(x_{kc_{j-1}} + x_{kc_{j+1}}) \le \frac{q-1}{2} + \sum_{v \in K \setminus \{k\}} z_{kv}. \tag{17}$$

On the other hand, we can write the LHS of (16) as

$$\sum_{c_j \in P}(x_{kc_{j-1}} + x_{kc_j} + x_{kc_{j+1}}) + \sum_{c_j \in P} \delta_{c_j}. \tag{18}$$

Hence we can prove that (16) holds by showing that (18) is not greater than the LHS of (17). To this end, for every $c_j \in P$ we show

$$(x_{kc_{j-1}} + x_{kc_j} + x_{kc_{j+1}}) + \delta_{c_j} \le 1 + \delta_{c_j}(x_{kc_{j-1}} + x_{kc_{j+1}}). \tag{19}$$

If $\delta_{c_j} = 0$, then (19) holds as $(x_{kc_{j-1}} + x_{kc_j} + x_{kc_{j+1}}) \le 1$. If $\delta_{c_j} = 1$, then (19) holds only if $x_{kc_j} = 0$, and this is true as $\delta_{c_j} = 1$ and K is a clique. By summing (19) over $c_j \in P$, we obtain that (18) is lesser or equal than the LHS of (17), hence (16) holds.

Since the solution y is arbitrary, the inequality (16) is valid for $PS(G, C)$. □

Theorem 2 (*[15]*). *If $|C| > \chi(G)$ and $|C| \ge |K| + 5$, then the consecutive colors inner clique inequalities* (16) *are facet-defining for $PS(G, C)$.*

It is interesting to note that we can define the set $Q = \{c_1, \dots, c_q\}$ even when c_1 resp. c_q represents a color outside the limits of C, i.e., when c_2 is the first color resp. c_{q-1} is the last color from C. The inequalities obtained by omitting the variables associated to c_1 resp. c_q are still valid, and they also define facets of $PS(G, C)$.

To conclude this section, note that for any undominated integer solution $y = (x, z) \in PS(G, C)$ there are several consecutive colors inner clique inequalities satisfied with equality by y. To this end, take a vertex $k \in V$ and call c the color assigned to it. If $c \ne 1, t$, then by defining $Q = \{c-2, c-1, c, c+1, c+2\}$ and taking any clique K such that $k \in K$, the associated valid inequality is satisfied with equality by y. If c is the first color resp. the last color of C, by defining $Q = \{c, c+1, c+2\}$ resp. $Q = \{c-2, c-1, c\}$ we again obtain a consecutive colors inner clique inequality satisfied with equality by y.

3.3. Consecutive colors subset clique inequalities

The class of valid inequalities introduced in this section arises from similar considerations as for the inner clique inequalities.

Definition 5. Let $K \subseteq V$ be a clique of G, and fix a vertex $k \in K$. Let $Q = \{c_1, \dots, c_q\} \subseteq C$ be a set of consecutive colors such that $c_{i+1} = c_i + 1$ for $i = 1, \dots, q-1$. We define the *consecutive colors subset clique inequality* associated with K, k and Q to be

$$\left(x_{kc_1} + 2x_{kc_2} + \sum_{i=3}^{q-2} 3x_{kc_i} + 2x_{kc_{q-1}} + x_{kc_q}\right) + \sum_{v \in K \setminus \{k\}} \sum_{i=2}^{q-1} x_{vc_i} \le (q-2) + \sum_{v \in K \setminus \{k\}} z_{vk}. \tag{20}$$

Proposition 5. *The consecutive colors subset clique inequalities* (20) *are valid for $PS(G, C)$.*

Proof. Let $y = (x, z) \in PS(G, C)$ be a feasible solution. For $c \in C$, define $\delta_c \in \{0, 1\}$ to be

$$\delta_c = \sum_{v \in K \setminus \{k\}} x_{vc},$$

i.e., $\delta_c = 1$ if and only if c is assigned to a vertex from $K \setminus \{k\}$. For every $i = 2, \dots, q-1$, note that if $\delta_{c_i}(x_{kc_{i-1}} + x_{kc_{i+1}}) = 1$, then a color adjacency exists between k and some vertex from $K \setminus \{k\}$. So, we can bound the RHS of (20) by

$$(q-2) + \sum_{i=2}^{q-1} \delta_{c_i}(x_{kc_{i-1}} + x_{kc_{i+1}}) \le (q-2) + \sum_{v \in K \setminus \{k\}} z_{kv}. \tag{21}$$

Note that no color adjacency is counted more than once since k receives at most one color from Q. On the other hand, we can write the LHS of (20) as

$$\sum_{i=2}^{q-1}(x_{kc_{i-1}} + x_{kc_i} + x_{kc_{i+1}}) + \sum_{i=2}^{q-1} \delta_{c_i}. \tag{22}$$

To conclude the proof, we show that (20) holds by verifying that (22) is not greater than the LHS of (21). To this end, note that

$$(x_{kc_{i-1}} + x_{kc_i} + x_{kc_{i+1}}) + \delta_{c_i} \leq 1 + \delta_{c_i}(x_{kc_{i-1}} + x_{kc_{i+1}}) \tag{23}$$

for every $i = 2, \ldots, q-1$. Indeed, if $\delta_{c_i} = 0$, then (23) holds as $(x_{kc_{i-1}} + x_{kc_i} + x_{kc_{i+1}}) \leq 1$. If $\delta_{c_i} = 1$, then (23) holds only if $x_{kc_i} = 0$, and this is true as $\delta_{c_i} = 1$. By summing (23) over $i = 2, \ldots, q-1$, and combining (22) and (21), we get the original valid inequality (20).

Since the solution y is arbitrary, the inequality (20) is valid for $PS(G, C)$. □

Theorem 3 (*[15]*). *If $|C| > \chi(G)$ and $|C| \geq |K| + 5$, the consecutive colors subset clique inequalities* (20) *are facet-defining for $PS(G, C)$.*

As in the consecutive colors inner clique inequalities, the set $Q = \{c_1, \ldots, c_q\}$ can be defined even when c_1 resp. c_q represents a color outside the limits of C. The inequalities obtained by omitting the variables associated to c_1 resp. c_q are still valid, and they also define facets of $PS(G, C)$.

Again, we conclude this section noting that for any undominated integer solution $y = (x, z) \in PS(G, C)$ there are several consecutive colors subset clique inequalities satisfied with equality by y. To this end, take a vertex $k \in V$ and call c the color assigned to it. If $c \neq 1, t$, then by defining $Q = \{c-2, c-1, c, c+1, c+2\}$ and taking any clique K such that $k \in K$, the associated valid inequality is satisfied with equality by y. If c is the first color resp. the last color of C, by defining $Q = \{c, c+1, c+2\}$ resp. $Q = \{c-2, c-1, c\}$ we obtain a consecutive colors subset clique inequality satisfied with equality by y (these are actually consecutive colors inner clique inequalities).

4. The branch-and-cut algorithm

We briefly describe in this section the implementation of a branch-and-cut algorithm for the minimum-adjacency vertex coloring problem, based on the polyhedral results presented in Section 3.

4.1. Formulation and valid inequalities

We start with the stable-set formulation (1)–(5), replacing the inequalities (3) by the dominating consecutive colors clique inequalities associated with each edge of G and each pair of consecutive colors. In the branch-and-cut algorithm we also consider the clique inequalities introduced in [7] in a similar context. If $K \subseteq V$ is a clique of G and $c \in C$, the *clique inequality* associated with K and c is

$$\sum_{v \in K} x_{vc} \leq 1. \tag{24}$$

This inequality is valid and, if K is a maximal clique, then (24) induces a facet of $PS(G, C)$. In our implementation we dynamically add cuts generated from the following classes of valid inequalities:

- consecutive colors inner clique inequalities (CCIK),
- consecutive colors subset clique inequalities (CCSK),
- multi-consecutive colors clique inequalities (MCCK), and
- clique inequalities (K).

4.2. Separation procedures

We have designed both exact and heuristic separation procedures for each class of valid inequalities, in order to address the tradeoff between the number of generated cuts and the separation times. As all these classes require a clique in the interference graph, we resort to generic clique-searching exact and heuristic methods for the CCIK, CCSK, and K inequalities, and the MCCK inequalities are handled in a slightly different way. We now describe these computational procedures.

Let $\hat{y} = (\hat{x}, \hat{z})$ be the fractional solution after solving the linear relaxation associated with some node in the branch-and-cut tree, and suppose we want to separate the CCIK inequalities, i.e., we search for a clique $K \subseteq V$, a vertex $k \in K$ and a set of consecutive colors $Q = \{c_1, \ldots, c_q\} \subseteq C$ such that

$$\left(\hat{x}_{kc_1} + \hat{x}_{kc_q} + \sum_{c_i \in I \setminus \{c_1, c_q\}} 2\hat{x}_{kc_i} + \sum_{c_j \in P} \hat{x}_{kc_j} \right) + \sum_{v \in K \setminus \{k\}} \sum_{c_j \in P} \hat{x}_{vc_j} > \frac{q-1}{2} + \sum_{v \in K \setminus \{k\}} \hat{z}_{vk}. \tag{25}$$

As the number of vertices in G and the number of odd-sized consecutive colors sets are both linear on the instance size, we search for a clique $K \subseteq V$ satisfying (25) for each possible combination of k and Q. Fix, therefore, some vertex $k \in V$ and some set $Q = \{c_1, \ldots, c_q\} \subseteq C$ and, for each vertex $v \in \mathcal{N}(k)$, define the *weight in $\hat{y}$* of v with respect to k as

$$\omega_k(v) = \left(\sum_{c_j \in P} \hat{x}_{vc_j} \right) - \hat{z}_{vk}.$$

Under this definition, (25) can be written as

$$\sum_{v\in K\setminus\{k\}} \omega_k(v) > \frac{q-1}{2} - \left(\hat{x}_{kc_1} + \hat{x}_{kc_q} + \sum_{c_i\in I\setminus\{c_1,c_q\}} 2\hat{x}_{kc_i} + \sum_{c_j\in P}\hat{x}_{kc_j}\right) = \hat{\sigma}_k$$

so the separation problem is, in this case, equivalent to searching for a clique K with weight greater than $\hat{\sigma}_k$. A similar approach can be performed for the CCSK and the K inequalities, by defining suitable vertex weights as follows:

$$\text{CCSK}: \omega_k(v) = \left(\sum_{i=2}^{q-1} \hat{x}_{vc_i}\right) - \hat{z}_{kv}$$

$$\text{K}: \omega_k(v) = \hat{x}_{vc}.$$

The exact clique-searching procedure is based on a backtracking algorithm, which searches for all possible cliques in the neighborhood of the vertex k. At each level in the backtracking tree, a new vertex from $\mathcal{N}(k)$ is considered for addition into/exclusion from the clique under construction. The implemented algorithm can be configured to yield (a) all cliques, (b) the first N, or (c) the best N cliques generating violated inequalities. As this procedure is clearly exponential, we have implemented the following additional techniques in order to make the search as efficient as possible:

- *Positive weights only*. We exclude from the search the vertices in $\mathcal{N}(k)$ with negative weights (note that negative weights are indeed possible in this setting).
- *Node limit*. We impose a limit on the number of nodes that the backtracking visits in the enumeration tree, in order to keep the separation times under control.
- *Node bounding*. At each node a in the backtracking tree, a bounding step is performed as follows. Let K be the partial clique associated with the node, and let T be the set of vertices not yet visited by the procedure (i.e., corresponding to the vertices associated with the subtree rooted at the node a). An upper bound of the best possible clique in the subtree rooted at the node a is given by $\omega(K) + \omega(T)$, attained if $K \cup T$ is indeed a clique. If this bound is smaller than the best clique found so far, then the node a is closed and its descendants are no further examined. It is interesting to note that $\omega(T)$ need not be computed at each node, since T corresponds to the last $|T|$ vertices of $\mathcal{N}(k)$, thus it only depends on the level of a within the tree. This way, only $|\mathcal{N}(k)| - 1$ bounds must be precomputed before the backtracking begins.
- *Vertex ordering*. The backtracking algorithm tests the vertices with larger weights first, in order to construct the first cliques in a greedy-like way, thus giving the bounding step more chances of succeeding.

These additions to the backtracking procedure allowed to cut significant portions of the enumeration tree, generating shorter running times without affecting the separation results.

On the other hand, the heuristic separation procedure consists of a standard greedy algorithm for clique searching. The algorithm starts with an arbitrary vertex $w \in \mathcal{N}(k)$ defining a singleton clique $K = \{w\}$ and visits the vertices of $\mathcal{N}(k) \setminus \{w\}$ in decreasing order of their weights, greedily adding each vertex v to K if $K \cup \{v\}$ is a clique.

4.3. Separation of the MCCK inequalities

The separation procedure for the MCCK inequalities first tries to find a violated consecutive colors clique (CCK) inequality, and then it tries to extend this inequality in order to find one or more violated MCCK inequalities.

The separation of the CCK inequalities consists in finding a clique K and a set of consecutive colors Q such that the CCK inequality associated with K and Q is violated. Since the total number of sets with consecutive colors is $O(t^2)$, we search for a clique K for every possible set Q of consecutive colors. Once the set Q is fixed, we define the *weight in* $\hat{y} = (\hat{x}, \hat{z})$ of $v \in V$ with respect to K as

$$\omega_K(v) = \left[\hat{x}_{vc_1} + \hat{x}_{vc_q} + \sum_{c\in Q\setminus\{c_1,c_q\}} 2\hat{x}_{vc}\right] - \sum_{w\in K}\frac{1}{2}\hat{z}_{vw}.$$

Note that, unlike the separation procedures in Section 4.2, in this case the weight depends on the clique K being constructed. Under this definition, a violated CCK inequality can be written as $\sum_{v\in K}\omega_K(v) > |Q| - 1$. The backtracking procedure is applied with these weights, but the vertex ordering and the bounding step cannot be applied in this case. To overcome this difficulty, for each vertex $v \in \mathcal{N}(k)$, we define the *estimated weight* to be

$$\tilde{\omega}(v) = \hat{x}_{vc_1} + \hat{x}_{vc_q} + \sum_{c\in Q\setminus\{c_1,c_q\}} 2\hat{x}_{vc}, \tag{26}$$

and such weights are used for ordering the vertices and perform the bounding step in the backtracking procedure. Note that these estimated weights do not depend on the clique K under construction. Since $\tilde{\omega}(v) \geq \omega_K(v)$ for every $v \in \mathcal{N}(k)$, these estimated weights can be properly used in order to perform the bounding step within the backtracking procedure. These estimated weights are also used in the greedy separation heuristic for the CCK inequalities.

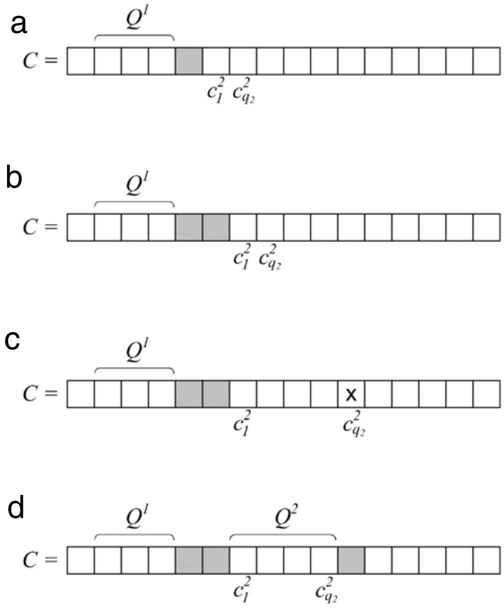

Fig. 2. Extension from a CCK to an MCCK inequality.

Once a violated CCK inequality is found, say associated with a clique $K \subseteq V$ and a set of consecutive colors $Q \subseteq C$, we try to extend this inequality to a violated MCCK inequality. To this end, we set $Q^1 = Q$ and we try to find new sets $Q^2, \ldots, Q^p \subseteq C$ defining a violated MCCK inequality.

We first search for a set $Q^2 = \{c_1^2, \ldots, c_{q_2}^2\}$ nonadjacent to Q^1, as in Fig. 2(a). If the associated MCCK inequality is not violated, both colors c_1^2 and $c_{q_2}^2$ are moved forward, as in Fig. 2(b), until the resulting inequality is violated. Once such an inequality is found, we try to enlarge the interval Q^2 by moving $c_{q_2}^2$ forward as long as the resulting inequality remains violated. When the inequality is not violated (as in Fig. 2(c)), we fix $c_{q_2}^2$ to be the preceding color (i.e., such that the resulting inequality is violated), so that Q^2 is set as in Fig. 2(d). We repeat this procedure until there are no colors left.

4.4. Additional techniques

Primal heuristic. We implement a standard primal heuristic based on solution rounding, in order to exploit the potential structure in the fractional solutions at each node in the branch-and-cut tree. Let $(\hat{x}, \hat{z})$ be the current (fractional) solution. We resort to a straightforward constructive heuristic, by selecting at each step a vertex v and a color c such that $\hat{x}_{vc} = \max\{\hat{x}_{v'c'} : 0 < \hat{x}_{v'c'} < 1\}$, and we set $\hat{x}_{vc} = 1$, $\hat{x}_{vc'} = 0$ for every $c' \in C \setminus \{c\}$, and $\hat{x}_{wc} = 0$ for every $w \in \mathcal{N}(v)$. These selections are performed in an arbitrary order and the LP problem is not resolved after each variable setting. This procedure is repeated until there are no x-variables with fractional values and, upon termination, the z-variables are given values accordingly. A feasible solution arises from this procedure if every vertex is assigned exactly one color.

Selection of the branching variable. Since the x-variables define the feasible solution, we branch on these variables only. When a branching step is performed, a fractional x-variable is selected and two children nodes are created, by setting this variable to 0 or 1, respectively.

Variable fixing by logical implications. Whenever a variable is fixed by the branching process, we perform the following straightforward variable fixings in order to reduce the subproblem size. For each $v \in V$ and $c \in C$ such that x_{vc} is fixed to 1, we set $x_{vc'} = 0$ for every $c' \in C \setminus \{c\}$ and $x_{wc} = 0$ for every $w \in \mathcal{N}(v)$. Furthermore, for every $vw \in E$ and $c_1, c_2 \in C$ with x_{vc_1} and x_{wc_2} both fixed to 1, we set $z_{vw} = 1$ if $|c_1 - c_2| = 1$ and $z_{vw} = 0$ otherwise.

5. Computational experiments

We tested the branch-and-cut algorithm on subgraphs from the CELAR instances set from the EUCLID CALMA project [16]. Although these instances do not explicitly define co-channel and adjacent-channel interference properties, we are interested in their real-world interference graphs. Since we are interested in the running times needed to achieve optimality, we randomly extract connected induced subgraphs from these instances in order to manage the size and density in our tests. To this end, we implemented a straightforward algorithm which starts from a random vertex and builds a subgraph by

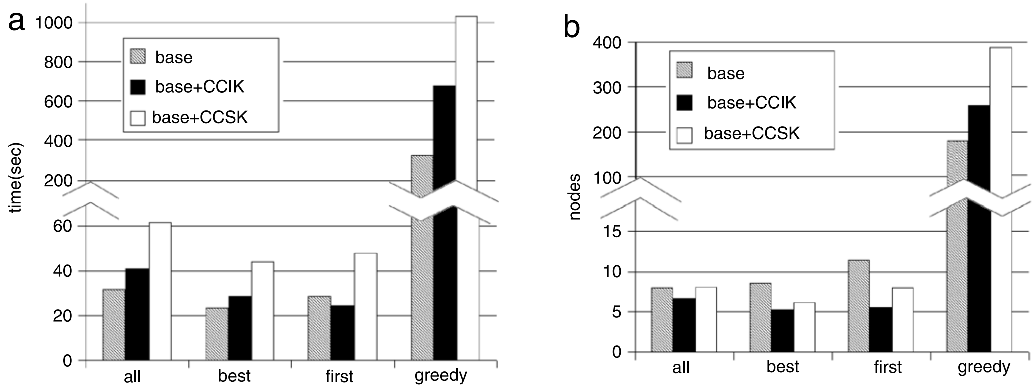

Fig. 3. Average (a) execution time in seconds and (b) number of nodes in the branch-and-cut tree for combinations of families of valid inequalities.

randomly selecting neighbors of the already selected vertices. This way we can control the instance size, while keeping graph structures present in real-world instances. We have considered instances with 14–20 vertices and, for the generated instances, we use between 9 and 13 channels.

We employed the ABACUS framework [17] to develop the branch-and-cut algorithm, linked to ILOG CPLEX 9.0 [18] for solving the linear relaxations. The experiments were performed on an AMD Athlon© 64 PC, with 1.5 GHz and a RAM memory of 2 GB, and a time limit of 30 min.

Selection of families of inequalities. We executed the branch-and-cut algorithm with several combinations of families of valid inequalities alone, resorting to the following separation procedures:

- backtracking procedure with the `first` 10 option,
- backtracking procedure with the `best` 10 option,
- backtracking procedure with the `all` option,
- greedy heuristic returning at most 10 cliques.

None of the considered families alone is able to obtain good results within the imposed time limit. In particular, the algorithm with just one of each family obtains no optimal solutions for instances with more than 14 vertices.

The *minimal* combination of valid inequalities achieving reasonable results is the union of the clique inequalities (K) and the multi-consecutive colors clique inequalities (MCCK). Using this combination, most instances with up to 20 vertices can be solved in less than 2 min, so the combination MCCK+K is taken as the *base configuration* for the following experiments. It is interesting that the combined action of both inequalities allows for such good results, as the K inequalities enforce the definition of a proper coloring (since they do not involve the adjacency variables) whereas the MCCK inequalities concentrate on the proper definition of the variables taking part in the objective function.

Starting with the base configuration, we study the effect of the remaining families in the overall procedure. In particular, we now report on the results with the following combinations:

- MCCK +K
- MCCK +K +CCIK
- MCCK +K +CCSK.

Fig. 3 reports the average solution time and nodes in the branch-and-cut tree for 30 instances between 16 and 20 vertices. A remarkable result is that the performance of the greedy separation procedure is not as effective as the backtracking-based separation for these instances. Both the running times and the number of nodes in the branch-and-cut tree are much larger when the greedy separation is applied, in spite of the shorter separation times. The best combination turns out to be the base configuration (MCCK + K inequalities) with a backtracking separation procedure returning the best 10 found cliques. However, the differences between this configuration and the other combinations of valid inequalities and backtracking options seem to be small.

Separation parameters. We now refine the experiments with the separation procedures for the MCCK + K inequalities, by tuning the parameters of the backtracking separation procedure returning the best found cliques. We experiment with the number N of cliques returned by the procedure ranging between 1 and 34, and with the maximum number of nodes in the backtracking trees ranging in the set $\{150, 300, 450, 600\}$. These experiments aim at assessing the tradeoff between the number of generated cuts and the separation times.

Fig. 4 reports the average results. The best running times are obtained when the number N of returned cliques ranges between 12 and 22, and when the backtracking procedure stops after 150 nodes in the backtracking tree. This result is interesting, as it suggests that the best cuts are obtained within the first branches of the backtracking tree and it is not worth visiting the remaining nodes in the backtracking algorithm.

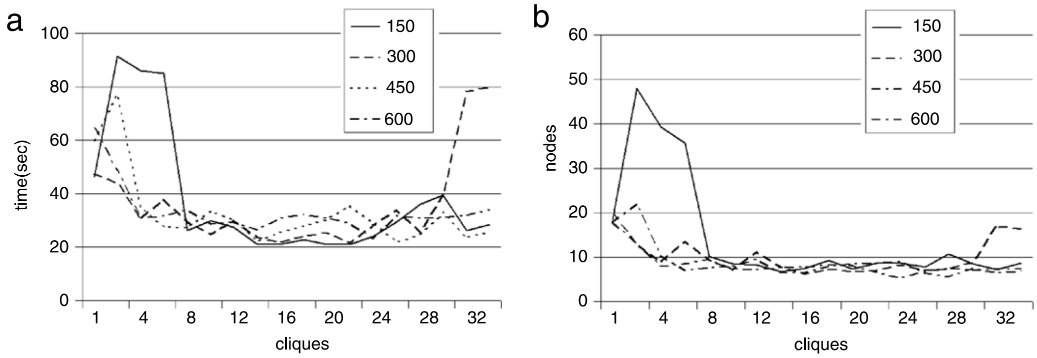

Fig. 4. Average (a) execution time in seconds and (b) number of nodes in the branch-and-cut tree for different combinations of the separation parameters.

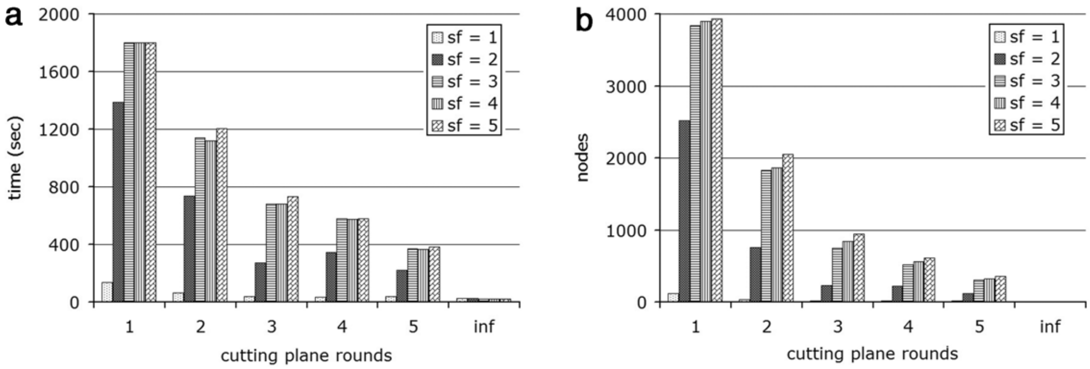

Fig. 5. Average (a) execution time in seconds and (b) number of nodes in the branch-and-cut tree for different combinations skip factor and number of cutting plane rounds.

Parameters of the branch-and-cut algorithm. In this paragraph we consider two key parameters of the branch-and-cut procedure, namely the *skip factor* and the number of *cutting plane rounds* at each node in the branch-and-cut tree. In our experiments the skip factor ranges within the set $\{1, \ldots, 5\}$ and the number of cutting plane rounds ranges within the set $\{1, \ldots, 5, \infty\}$.

Fig. 5 reports the average results over 30 instances. It is observed that as the number of cutting plane rounds increases, the overall performance is better (e.g., with just one cutting plane round and skip factors of 3, 4, and 5, no instance is solved to optimality), the best value being "infinite". The influence of the skip factor is also remarkable, showing the best results when the cutting plane phase is applied at every node in the branch-and-cut tree (i.e., with skip factor equal to 1). These results suggest to give priority to the cutting plane phase over the branching steps, giving a hint that the combination MCCK + K is indeed strong.

Comparison with CPLEX [18]. We tested our implementation with the best parameter setting found in the previous paragraphs, comparing against the running times obtained by CPLEX 9.0. Table 1 shows the running times in seconds and the number of nodes in the enumeration tree for our implementation and CPLEX. For the instances that could not be solved to optimality within the time limit, we also report the final duality gap. For the branch-and-cut algorithm, the number of K and MCCK generated cuts and the overall separation time are also reported, as well as the dual bound achieved by the cutting plane phase at the root node of the tree. Note that the reported gap at this column is computed against the optimal solution for the instance. It is interesting to note that the optimal value of the initial linear relaxation is always zero.

It is interesting to note that, for the considered instances, both the running times and the number of generated nodes are much smaller for the branch-and-cut algorithm. Furthermore, only one instance is not solved to optimality by the branch-and-cut while CPLEX fails to achieve optimality in 80% of the instances. The gap obtained in this unsolved instance is the same for both algorithms while the number of open nodes is significantly smaller in the branch-and-cut. Another interesting result relates to the dual bounds obtained by applying the cuts at the root node. In particular, in 92.5% of the instances the dual bound achieved at the root node equals the optimal value (recall that without the cut generation, the linear relaxation has null optimal value).

These results show that the dynamical generation of the MCCK inequalities and the K inequalities are indeed a good addition to the basic branch-and-bound algorithm, resulting in much better running times and lower bounds.

Table 1
Comparison between the branch-and-cut algorithm with the best parameter setting and CPLEX 9. Times are reported in seconds.

$\|V\|$	Density (%)	*Branch & Cut*						CPLEX		
		K	MCCK	Sep. time	Dual at root/opt. val. (gap (%))	Nodes	Time	Time	Nodes	Gap (%)
14	48.35	169	1196	1.02	5/5 (0)	**3**	**1.91**	**240**	**465.100**	
	49.45	116	553	0.93	2/2 (0)	**5**	**1.72**	**16**	**21.000**	
	49.45	137	676	1.47	5/5 (0)	**5**	**2.42**	**380**	**701.800**	
	56.04	242	1139	1.2	9/9 (0)	**5**	**2.61**	***	**5.878.200**	33
	57.14	229	1024	2.38	6/6 (0)	**9**	**4.26**	**108**	**124.500**	
	58.24	273	1329	3.39	6/7 (17)	**11**	**6.39**	**188**	**210.700**	
	59.34	279	1531	0.73	7/7 (0)	**1**	**2.1**	**149**	**118.800**	
	60.44	224	1317	3.64	4/4 (0)	**13**	**7.03**	**63**	**59.800**	
	64.84	640	6728	679.89	8/9 (14)	**2331**	**1714.81**	**1650**	**1.699.800**	
	65.93	497	7145	662.9	–	**2593**	***/9%	***	**3.766.100**	9
16	55.00	227	1490	2.53	6/6 (0)	**5**	**5.18**	***	**3.596.400**	58
	56.67	156	1477	2.12	6/6 (0)	**3**	**4.09**	***	**2.783.800**	22
	56.67	311	1900	5.39	9/9 (0)	**15**	**12.04**	***	**3.273.400**	69
	60.00	288	1814	1.7	8/8 (0)	**1**	**4.66**	***	**1.969.900**	45
	60.00	288	1814	1.7	8/8 (0)	**1**	**4.63**	***	**1.975.800**	45
	60.00	288	1814	1.75	8/8 (0)	**1**	**4.73**	***	**1.978.600**	45
	60.00	288	1814	1.75	8/8 (0)	**1**	**4.68**	***	**1.972.400**	45
	60.83	453	2641	12.52	8/8 (0)	**17**	**26.81**	***	**2.487.200**	56
	62.50	333	2580	2.2	9/9 (0)	**1**	**10.55**	***	**2.081.400**	48
	70.00	468	2510	5.04	11/11 (0)	**3**	**19.27**	***	**1.041.500**	61
18	46.41	337	2053	4.95	8/8 (0)	**3**	**8.88**	***	**2.674.800**	72
	47.06	180	1669	4.47	6/6 (0)	**3**	**7.22**	***	**2.841.700**	67
	54.25	312	2133	3.14	6/6 (0)	**3**	**6.53**	***	**1.610.200**	75
	56.21	296	2090	1.76	10/10 (0)	**1**	**6.13**	***	**1.645.600**	76
	57.52	557	2395	2.94	12/12 (0)	**1**	**8.96**	***	**1.235.100**	75
	60.13	378	2803	7.01	10/10 (0)	**5**	**20.15**	***	**1.508.400**	75
	61.44	180	1620	1.44	12/12 (0)	**1**	**5.14**	***	**1.250.000**	75
	64.05	568	4344	11.25	12/12 (0)	**5**	**45.8**	***	**1.032.900**	75
	66.67	452	3653	4.37	11/11 (0)	**1**	**31.51**	***	**1.239.600**	81
	73.86	888	3501	18.13	12/12 (0)	**9**	**53.57**	***	**572.100**	79
20	38.95	193	2806	7.12	5/5 (0)	**11**	**17.73**	***	**3.693.200**	75
	41.05	265	2952	9.7	5/5 (0)	**13**	**23.02**	***	**3.138.800**	80
	46.84	309	2626	2.09	8/8 (0)	**1**	**7.98**	***	**1.897.300**	75
	47.37	226	1747	3.32	2/2 (0)	**3**	**7.42**	***	**1.454.500**	50
	48.42	224	2594	10.54	6/6 (0)	**13**	**22.65**	***	**1.926.700**	72
	50.00	198	2938	15.13	6/6 (0)	**17**	**34.34**	***	**1.329.000**	75
	53.16	284	2883	12.04	4/4 (0)	**9**	**26.36**	***	**885.800**	75
	53.16	312	2649	8.87	4/4 (0)	**7**	**19.31**	***	**800.500**	75
	54.21	295	2723	9.29	4/4 (0)	**11**	**26.56**	***	**957.200**	75
	68.42	972	5354	10.32	13/13 (0)	**1**	**103.74**	***	**572.300**	77

6. Conclusions and future work

In this work we studied the minimum-adjacency vertex coloring problem, which captures the combinatorial structure of a natural way of dealing with interference in many wireless communications networks. We have performed a polyhedral study of an integer programming formulation for this problem, presenting three facet-inducing families of valid inequalities. We have designed separation procedures for these families and implemented a branch-and-cut algorithm based on these results, which obtained competitive results with respect to the existing computational machinery. Although the instances we are able to solve to optimality are far from being real-size instances, we believe that the results presented in this work may contribute to future developments for the practical solution of frequency assignment problems.

It would be interesting to search for further families of (facet-inducing) valid inequalities, in order to enhance the computational results obtained in this work and, in particular, in order to tackle larger problem instances. Our experiments suggest that not any class of valid inequalities may benefit the overall branch-and-cut algorithm, hence such a polyhedral study should be accompanied by computational experiments in order to assess the practical contribution of each family in such an environment.

Acknowledgments

We thank the anonymous referees for their comments and suggestions, which greatly helped to improve this work. In particular, we are grateful with one referee's remark which allowed us to find the classes of valid inequalities (16) and (20), which generalize our initial results.

The authors were partially supported by UBACYT Grant X069 and ANPCyT PICT-2007-00518.

References

[1] A. Eisenblätter, Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds, Ph.D. Thesis, Technische Universität Berlin, 2001.
[2] A. Eisenblätter, Assigning frequencies in GSM networks, Operations Research Proceedings 2002 (2003) 33–40.
[3] M. Grötschel, Frequency Assignment in Mobile Phone Systems, in: LNCS, vol. 1974, 2000, pp. 81–86.
[4] F. Luna, E. Alba, A. Nebro, S. Pedraza, Evolutionary Algorithms for Real-World Instances of the Automatic Frequency Planning Problem in GSM Networks, in: Lecture Notes in Computer Science, vol. 4446, 2007, pp. 108–120.
[5] K. Aardal, S. Van Hoesel, A. Koster, C. Mannino, A. Sassano, Models and solution techniques for frequency assignment problems, Annals of Operation Research 153 (2007) 79–129.
[6] A. Eisenblätter, The Semidefinite Relaxation of the k-Partition Polytope is Strong, in: Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization, 2002, pp. 273–290.
[7] I. Méndez-Díaz, P. Zabala, A branch-and-cut algorithm for graph coloring, Discrete Applied Mathematics 154 (5) (2006) 826–847.
[8] I. Méndez-Díaz, P. Zabala, A cutting plane algorithm for graph coloring, Discrete Applied Mathematics 156 (2) (2008) 159–179.
[9] R. Borndörfer, A. Eisenblätter, M. Grötschel, A. Martin, The Orientation Model for Frequency Assignment Problems, ZIB-Berlin TR 98-01, 1998.
[10] M. Campêlo, R. Corrêa, Y. Frota, Cliques, holes and the vertex coloring polytope, Information Processing Letters 89 (4) (2004) 159–164.
[11] A. Mehrotra, M. Trick, A column generation approach for graph coloring, INFORMS Journal On Computing 8 (4) (1996) 344–354.
[12] T.D. Hemazro, B. Jaumard, O. Marcotte, A column generation and branch-and-cut algorithm for the channel assignment problem, Computers & Operations Research 35 (4) (2008) 1204–1226.
[13] B. Jaumard, O. Marcotte, C. Meyer, T. Vovor, Comparison of column generation models for channel assignment in cellular networks, Discrete Applied Mathematics 112 (2001) 217–240; Discrete Applied Mathematics 118 (3) (2002) 299–322 (erratum).
[14] D. Delle Donne, A branch-and-cut algorithm for a frequency assignment problem in celular phones networks, Graduate Thesis, University of Buenos Aires, 2009.
[15] D. Delle Donne, J. Marenco, Facets of the minimum-adjacency vertex coloring polytope, Technical Report, National University of General Sarmiento, 2010. Available at http://www.optimization-online.org/DB_HTML/2010/10/2750.html.
[16] A. Eisenblätter, A. Koster, FAPWeb — a frequency assignment website, 2009. http://fap.zib.de.
[17] M. Jünger, S. Thienel, The ABACUS system for branch-and-cut-and-price algorithms in integer programming and combinatorial optimization, Software — Practice & Experience 30 (11) (2000) 1325–1352.
[18] IBM ILOG, User's Manual for CPLEX, 2009.