Directed weighted improper coloring for cellular channel allocation

Claudia Archetti[(1)] *Nicola Bianchessi*[(2)]
Alain Hertz[(3)*] *Adrien Colombet*[(4)] *François Gagnon*[(4)]

[(1)] *Department of Quantitative Methods*
University of Brescia, Brescia, Italy

[(2)] *Department of Information Engineering*
University of Brescia, Brescia, Italy

[(3)] *Department of Mathematics and Industrial Engineering*
École Polytechnique and GERAD, Montréal, Canada

[(4)] *Department of Electrical Engineering*
École de Technologie Supérieure, Montréal, Canada

November 11, 2013

Abstract

Given a directed graph with weights on the vertices and on the arcs, a θ-improper k-coloring is an assignment of at most k different colors to the vertices of G such that the weight of every vertex v is greater, by a given factor $\frac{1}{\theta}$, than the sum of the weights on the arcs (u, v) entering v with the tail u of the same color as v. For a given real number θ, we consider the problem of determining the minimum integer k such that G has a θ-improper k-coloring. Also, for a given integer k, we consider the problem of determining the minimum real number θ such that G has a θ-improper k-coloring. We show that these two problems can be used to model channel allocation problems in wireless communication networks, when it is required that the power of the signal received at a base station is greater, by a given factor, than the sum of interfering powers received from mobiles which are assigned the same channel. We propose set partitioning formulations for both problems and describe branch-and-price algorithms to solve them. Computational experiments are reported for instances having a similar structure as real channel allocation problems.

Keywords: graph coloring, weighted directed graphs, channel assignment, set partitioning formulations, branch-and-price algorithm.

*Corresponding author : email alain.hertz@gerad.ca; tel. +1-514 340 6053; fax +1-514 340 5665.

1 Motivation

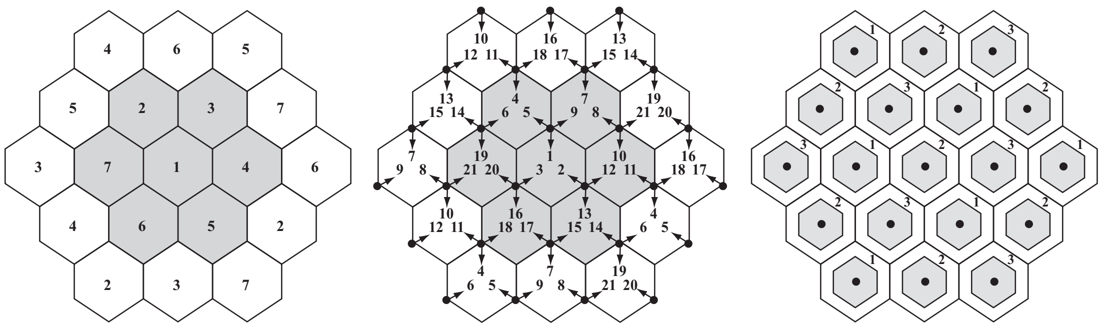
In wireless communication networks of the first and second generation, the concepts of cellular channel allocation and spatial frequency reuse were the key ideas that have driven the immense initial success of mobile telephony [13]. In this context, geographical regions were divided into cells, theoretically hexagonal, and each cell had a dedicated number of antennas with an associated set of frequency bands. For example, in the original AMPS, American Mobile Phone System, seven sets, which can be further subdivided into three subsets, were the original 'colors' with which one would allocate the channels to the regions (see Figures 1(a) and 1(b)). In actual systems, many configuration of base stations are deployed according to the geo-demographic situation. In simple systems and in low density demographic areas, the base stations use omni-directional antennas and are thus viewed as being at the centre of the cells 1(a). However in most systems, the base stations have three, 120° sectored, antennas, which permits to effectively discriminate the radio visibility horizon into three distinct parts. The base station is then viewed as being at the edges of three cells as in figure 1(b).

A similar scheme existed for GSM and the practical allocation problem consisted in selecting the set of channels associated with a cell or even sub-sets associated with an antenna. The criteria for channel selection consists in minimizing the mutual interference caused by transmission on the same channel in different cells or geographical regions. This may be reduced to maximizing the geographical distance between channel reuse. With real world cells, which differ in size and shape, the problem typically required coloring algorithms to solve them [18, 7].

In the third generation of mobile systems, the introduction of CDMA (Code Division Multiple Access) has enabled the reuse of the whole frequency band in each cell [13]. Instead of dividing the signal space in time or frequency between users, a code or pseudo-random sequence is used to differentiate the signal from each transmitter. Interference is tolerated up to a certain degree between transmitters within an area. In this context the coloring schemes and associated research projects were of much reduced importance.

In their latest incarnations, the fourth generation mobile standards mainly use Orthogonal Frequency Division Multiple Access schemes [12]. These schemes divide the signal space in time slots and orthogonal frequencies. At the middle of a cell, all slots of time and frequency are allocated to users. These cover the whole frequency band. At the edge of the cell, only part of the band is used and a three color scheme is used (see Figure 1(c)): only a third of the whole frequency band is available at the edge of each cell to reduce interference. In these latest standards the base station uses sectored but also multiple antennas, which are jointly managed. The base station is thus usually viewed as being in the centre of the cell. This is what we will assume in this paper. Furthermore, a form of spatial multiplexing may also be used within a cell by applying MIMO techniques (Multiple-Input Multiple-Output). This permits to allocate the same time-frequency slot to a number of users which are joined through quasi-independent paths. By storing the signal gain between each pair of antennas in a matrix, it is possible to perform some forms of diagonalization of this matrix to obtain independent transmission streams.

For these latest standards, the superposition of signals is tolerated to a certain degree [17]. That is, for a given signal that is denoted as the desired signal, one can tolerate the superposition of all other signals if their projection on the signal space-time is less powerful than the desired signal by a certain factor. Finally, when distances are relatively short, such as in dense urban areas, the random thermal noise, always present at a received antenna, becomes non-significant and the interference from other users or other cell sites dominate the communication performances. In this context, the capacity of the wireless communication scheme is said to be interference limited or dominated. The allocation of a time-frequency slot is akin to the allocation of a color. As a simplifying factor but without loss of generality, we will discount the angular separation of mobiles and compute the contribution of each mobile to the reception of the same colored signals at a base station as a function of distance.



(a) (b) (c)

Figure 1. Three different types of channel allocations.

More formally, let $M = \{m_1, \cdots, m_n\}$ be a set of mobiles and $B = \{b_1, \cdots, b_t\}$ a set of base stations. Let d_{ip} denote the Euclidian distance between the mobile m_i and the base station b_p. The positions are assumed here to be in two dimensions (in three dimensions one must account for a number of other factors such as floors). The received power P_{ip} at b_p of a signal from m_i is computed as follows:

$$P_{ip} = \frac{\alpha \, r_{ip}}{{d_{ip}}^{\gamma}}$$

where

- α is a constant summarizing the effects of antenna configuration and position,
- r_{ip} is a random variable synthesizing the desired channel models: it would typically be the product of an exponential random variable to denote short term fading and a log-normal random variable for shadowing [7],
- γ is the attenuation factor which typically varies from 2 in free-space to 4 for Non-Line Of Sight (NLOS) in urban areas.

Each mobile is assumed to be assigned to a given base station. Let $c(i)$ denote the channel, or color, assigned to mobile m_i. Also, for a channel k, let us denote by C_k the set of mobiles m_i assigned with channel $c(i) = k$. An admissible coloring scheme would require that the power of all received signals at their assigned base stations are greater, by a given factor $\frac{1}{\theta}$, than the sum of interfering powers received from mobiles which are assigned the same channel. More precisely, for a mobile m_i assigned to a base station b_p, we have the following constraint:

$$\sum_{m_j \in C_{c(i)}, j \neq i} P_{jp} \leq \theta \, P_{ip}. \tag{1}$$

where $\frac{1}{\theta}$ is the maximal admissible signal-to-interference ratio.

The rest of the paper is organized as follows. In the next section, we propose two graph coloring models which are equivalent to finding a channel assignment satisfying (1). Set partitioning formulations for both problems are given in Section 3, while branch-and-price algorithms are proposed in Section 4. Computational experiments are reported in Section 5 for randomly generated instances.

2 Graph coloring problems

Given an undirected graph $G = (V, E)$ with vertex set V and edge set E, a *k-coloring* of G is a function $c : V \to \{1, \cdots, k\}$. The coloring c is *proper* if no edge has both endpoints with the same color, otherwise it is *improper*. A k-coloring is θ-improper if every vertex v is adjacent to at most θ vertices having the same color as v. For a given graph G and an integer θ, the *Improper Coloring Problem* (ICP) (also called *Defective Coloring Problem* in [10]) is to determine the minimum integer k such that G has a θ-improper k-coloring. The ICP is NP-hard since it is proved in [10] that the problem of deciding if there is a θ-improper k-coloring of a graph G is NP-complete for $k = 2$ and $\theta \geq 1$ and for all pairs (θ, k) with $k \geq 3$ and $\theta \geq 0$.

In this paper, we study an extension of the ICP to directed weighted graphs. Let $G = (V, A, W, \omega)$ be a directed graph with vertex set V, arc set A, and with two functions

$W : V \to \mathbb{R}_+^*$ and $\omega : A \to \mathbb{R}_+^*$ that associate a weight $W(v)$ to every vertex $v \in V$ and a weight $\omega(u,v)$ to every arc $(u,v) \in A$. We denote N_v^- the set of immediate predecessors of vertex v (i.e., $u \in N_v^-$ if and only if $(u,v) \in A$). For a real number θ, we say that a function $c : V \to \{1, \cdots, k\}$ is a θ-improper k-coloring of G if, for every vertex $v \in V$, the following constraint is satisfied:

$$\sum_{u \in N_v^- \mid c(v)=c(u)} \omega(u,v) \leq \theta\, W(v). \tag{2}$$

For a given directed graph $G = (V, A, W, \omega)$ and a real number θ, we define the *Directed Weighted Improper Coloring Problem* (DWICP) which is to determine the minimum integer k such that G has a θ-improper k-coloring. The link between the channel assignment problem of Section 1 and the DWICP can now easily be seen. Indeed, for a set $M = \{m_1, \cdots, m_n\}$ of n mobiles and a set $B = \{b_1, \cdots, b_t\}$ of t base stations, we can construct a complete directed graph $G = (M, A, W, \omega)$ with an arc in both directions between each pair of mobiles, and with $W(m_i) = P_{ip}$ and $\omega(m_j, m_i) = P_{jp}$, where b_p is the base station to which m_i is assigned. The construction of G is illustrated in Figure 2(a). It follows from these weight definitions that equations (1) and (2) are equivalent, which means that finding a channel assignment satisfying (1) is equivalent to determining a θ-improper k-coloring of G.

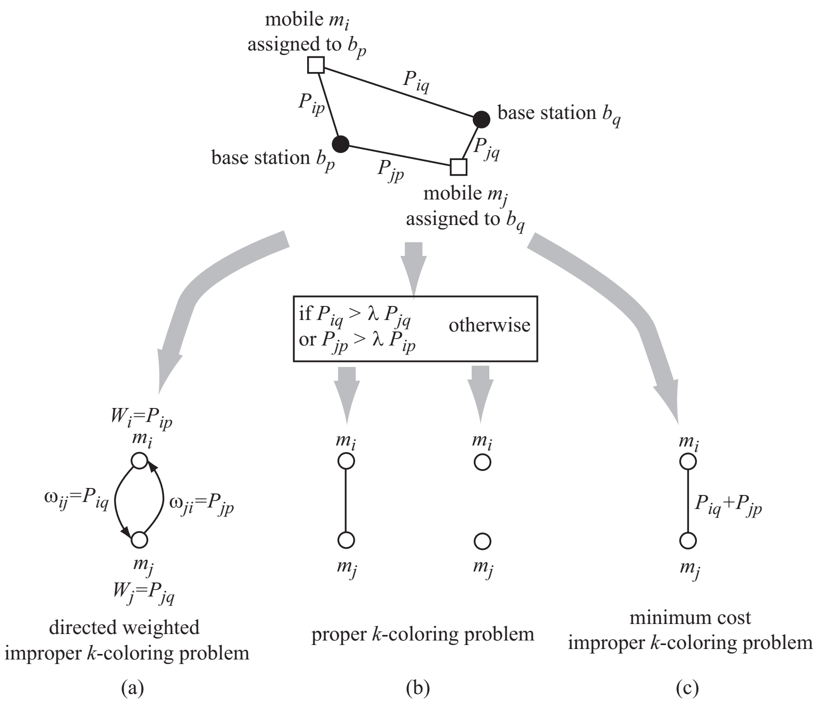


Figure 2. Three graph coloring models.

For a given directed graph $G = (V, A, W, \omega)$ and a positive integer k, we also define the *Directed Threshold Improper Coloring Problem* (DTICP) which is to determine the minimum real number θ such that G has a θ-improper k-coloring.

Araujo et al. [1] have studied the DWICP and the DTICP in the particular case where $W(v) = 1$ for all vertices v in G and $\omega(u, v) = \omega(v, u)$ for all pairs $\{u, v\}$ of vertices in G (which means that there is an arc from u to v if and only if there is an arc from v to u and the corresponding weights are identical). In such a case, G can be considered as undirected since each pair (u, v) and (v, u) of arcs can be replaced by an edge $\{u, v\}$ of weight $\omega(u, v)$ linking u with v. It is proved in [1] that both the WICP and TICP (i.e., the undirected versions of the DWICP and DTICP) are NP-hard, which means that both the DWICP and DTICP are also NP-hard.

Other graph coloring models have already been proposed for the solution of channel assignment problems. For example, instead of imposing constraints (1), some mobile operators prefer to avoid interference by forbidding the use of the same channel by mobiles m_i and m_j if P_{jp} is larger then $\lambda\, P_{ip}$ for a given real number λ, where b_p is the base station to which m_i is assigned. One would typically impose a minimal distance, counted in cells, with which a color is reused. Hence, for a mobile m_i assigned to a base station b_p, equation (2) is replaced by

$$P_{jp} \leq \lambda\, P_{ip} \qquad \forall\, m_j \in C_{c(i)}.$$

In such a case, one can simply build an undirected graph $G = (M, E)$ in which two mobiles m_i and m_j are linked by an edge if and only if $P_{jp} > \lambda\, P_{ip}$ or $P_{iq} > \lambda\, P_{jq}$, where b_p and b_q are the base stations to which m_i and m_j are assigned respectively. The channel assignment problem is then to determine a proper k-coloring in G, where k is the number of available channels. The construction of G is illustrated in Figure 2(b). Note that this can be seen as a special case of the DWICP, where θ is any real number strictly smaller than 1, $W(m_i) = 1$ for all vertices $m_i \in M$, and

$$\omega(m_i, m_j) = \omega(m_j, m_i) = \begin{cases} 1 \text{ if there is an edge in } G \text{ linking } m_i \text{ with } m_j \\ 0 \text{ otherwise.} \end{cases}$$

A variant could be to try to minimize interferences. Consider the complete undirected graph $G = (M, E)$ and let us associate a weight $\omega(m_i, m_j) = P_{jp} + P_{iq}$ to the edge linking m_i with m_j in G, where b_p and b_q are the base stations to which mobiles m_i and m_j are assigned respectively. The construction is illustrated in Figure 2(c). By denoting E_r the set of edges having both endpoints with color r, the channel assignment problem is then to find an improper k-coloring with minimum total interference computed as $\sum_{r=1}^{k} \sum_{\{m_i,m_j\} \in E_r} \omega(m_i, m_j)$. Such a coloring problem was studied, for example, in [16].

Another popular model for cellular channel allocation is based on $L(p, q)$ labelings. Given a graph G and two non-negative integers p and q, an $L(p, q)$-labeling of G is a labeling of its vertices such that adjacent vertices are labeled using colors at least p apart, and vertices having a common neighbor are labeled using colors at least q apart. It follows that an $L(1, 0)$-labeling of G is also a proper coloring of its vertices. For a survey of this widely studied problem the reader is referred to [9].

3 The problem formulations

Given a directed graph $G = (V, A, W, \omega)$, a real number $\bar{\theta}$ and positive integer $\bar{k}$, both the DWICP and DTICP can be modeled by means of set partitioning formulations. Let us consider the following notations:

- $P(V)$ is the set of all nonempty subsets of V;
- $\Omega_v^s = \sum_{u \in s} \omega_{uv}$ is the weight covered by vertex $v \in s$, $s \in P(V)$.

The DWICP aims at finding the minimum number of color classes k such that G has a $\bar{\theta}$-improper k-coloring for a given real number $\bar{\theta}$. We denote $S_{\bar{\theta}}$ the set of all subsets s of vertices such that $\Omega_v^s \leq \bar{\theta} W_v$ for all $v \in s$, i.e., $S_{\bar{\theta}}$ is the set of all feasible color classes w.r.t $\bar{\theta}$. The problem can be modeled as follows:

$$\min \sum_{s \in S_{\bar{\theta}}} \sigma_s \tag{3}$$

$$\sum_{s \in \mathcal{S}_{\bar{\theta}} \,|\, v \in s} \sigma_s = 1 \qquad \forall v \in V \tag{4}$$

$$\sigma_s \in \{0, 1\} \qquad \forall s \in \mathcal{S}_{\bar{\theta}}, \tag{5}$$

where σ_s is a binary variable equal to 1 if subset $s \in S_{\bar{\theta}}$ is selected as color class, 0 otherwise. The objective function (3) calls for the minimization of the number of classes used to color all the vertices. Constraints (4) impose to associate a color with each vertex. Constraints (5) ensure integrality requirements for all the variables.

On the other hand, the DTICP is to determine the minimum real number θ such that G has a θ-improper $\bar{k}$-coloring for a given positive integer $\bar{k}$. Here θ is a variable, and the actual value of θ defines the set of all feasible color classes. The problem model is the following:

$$\min \theta \tag{6}$$

$$\sum_{s \in P(V)\,|\,v \in s} \sigma_s = 1 \qquad \forall v \in V \tag{7}$$

$$\sum_{s \in P(V)\,|\,v \in s} \Omega_v^s \, \sigma_s \leq \theta \, W_v \qquad \forall v \in V \tag{8}$$

$$\sum_{s \in P(V)} \sigma_s = \bar{k} \tag{9}$$

$$\sigma_s \in \{0, 1\} \qquad \forall s \in P(V) \tag{10}$$

where σ_s is a binary variable equal to 1 if subset $s \in P(V)$ is selected as color class, 0 otherwise. The objective function (6) calls for the minimization of the θ-value guaranteeing the feasibility of the selected color classes. Constraints (7) impose the coloring of each vertex. Constraints (8) ensure the feasibility of the selected color classes according to θ. Constraints (9) fix the number of color classes to select. Finally, constraints (10) are the integrality constraints on the σ_s variables.

In the following, we will refer to problem (3)–(5) or (6)–(10) as the master problems (MPs); distinctions will be made whenever required.

4 The branch-and-price algorithms

The number of variables characterizing the MPs, i.e., the number of vertex subsets defining color classes, is exponential and, even for small size instances, solving the problems explicitly is impractical. We thus develop two branch-and-price algorithms ([8, 11]). For the considered MP, at each node of the branch-and-bound tree, variables (columns) of the problem are generated applying the column generation technique to the linear relaxation of the so called restricted MP, augmented by the branching constraints (see [11] for a complete survey of column generation methods). We denote the linear relaxation of the restricted MP, augmented by the branching constraints, as RLMP. LMP is the linear relaxation of MP augmented by the branching constraints. At each column generation iteration, a pricing problem, also called subproblem, is solved in order to generate negative reduced cost variables to be added to the RLMP. When no negative reduced cost variable is found, the LMP has been solved to optimality and the column generation algorithm ends. Branching rules are applied to recover feasibility when the solution of the LMP is fractional. In the following subsections, the main components of the algorithms addressing the MPs (3)–(5) and (6)–(10) will be illustrated in details, making distinctions whenever required.

4.1 Pricing problems

The pricing problems consist in finding the vertex subset s yielding the most negative reduced cost. More formally, let us consider the following binary variables:

- for every vertex $v \in V$, we define $x_v = 1$ if vertex v belongs to s, 0 otherwise;
- for every arc $(u, v) \in A$, we define $y_{vu} = 1$ if both vertices u and v belong to s, 0 otherwise.

While addressing the DWICP, at the root node of the branch-and-bound tree, the pricing problem is the following:

$$\min \quad \bar{c}^* = -\sum_{v \in V} \pi_v \, x_v \tag{11}$$

$$x_u + x_v - 1 \leq y_{uv} \qquad \forall (u, v) \in A \tag{12}$$

$$\sum_{u \in N_v^-} \omega_{uv} \, y_{uv} \leq \bar{\theta} \, W_v \, x_v \qquad \forall v \in V \tag{13}$$

$$x_v \in \{0, 1\} \qquad \forall v \in V \tag{14}$$

$$y_{uv} \in \{0, 1\} \qquad \forall (u, v) \in A \tag{15}$$

where π_v is the dual cost associated with constraint (4) for vertex v. Constraints (12) ensure consistency among the x and the y variables, while constraints (13) impose that the chosen vertex subset belongs to $S_{\bar{\theta}}$.

When the DTICP is considered, the pricing problem at the root node of the branch-and-bound tree is the following:

$$\min \quad \bar{c}^* = -\sum_{v \in V} \pi_v x_v - \sum_{v \in V} \rho_v \sum_{u \in N_v^-} \omega_{uv} y_{uv} - \beta \tag{16}$$

$$x_u + x_v - 1 \leq y_{uv} \qquad \forall (u, v) \in A \tag{17}$$

$$2y_{uv} - x_u - x_v \leq 0 \qquad \forall (u, v) \in A \tag{18}$$

$$x_v \in \{0, 1\} \qquad \forall v \in V \tag{19}$$

$$y_{uv} \in \{0, 1\} \qquad \forall (u, v) \in A \tag{20}$$

where:

- π_v is the dual cost associated with constraint (7) for vertex v;
- ρ_v is the dual cost associated with constraint (8) for vertex v;
- β is the dual cost associated with constraint (9).

Here, consistency among the x and the y variables is ensured by imposing both constraints (17) and (18). Moreover, in this case, the chosen vertex subset belongs to $P(V)$. If any upper bound $\bar{\bar{\theta}}$ on the optimal value of θ would be known in advance, the search space could be reduced by adding constraints

$$\sum_{u \in V} \omega_{uv} y_{uv} \leq \bar{\bar{\theta}} W_v x_v \quad \forall v \in V \tag{21}$$

to (17)–(20), and imposing thus to select the vertex subset in $S_{\bar{\bar{\theta}}}$.

At non-root nodes of the branch-and-bound tree, the pricing problem models are modified by taking into account branching constraints as described in Section 4.3.

The aim of solving the pricing problem is to generate variables of the LMP with a negative reduced cost. In order to try to avoid solving this problem with an exact method (which can be very time consuming), we first use the following heuristic pricing algorithm. Each column belonging to the optimal basis of the current RLMP is considered as the initial solution s (i.e., subset of vertices) of a descent method. We then evaluate (using function (11) or (16)) all feasible neighbors obtained from s by inserting a new vertex, removing a vertex, or replacing a vertex $u \in s$ by a vertex $v \notin s$. If the best feasible neighbor is strictly better than s, then the process is repeated with the best neighbor as new current solution s. All negative reduced cost columns found during this process are returned to the column generation algorithm.

If no negative reduced cost column is found by the heuristic pricing algorithm, we solve the pricing problem exactly by introducing model (11)–(15) or (16)–(21), eventually augmented by the branching constraints, in a commercial MILP solver.

4.2 The column generation algorithms

Let S be equal to $S_{\bar{\theta}}$ and $S_{\bar{\bar{\theta}}}$ while addressing the DWICP and the DTICP, respectively. The RLMP is initialized by means of a set $S' \subseteq S$ defined as follows. At the root node of the branch-and-bound tree, S' includes a high cost dummy column through which all the MP constraints are satisfied. This column is kept in the RLMP until feasibility is reached. Then, as far as the DWICP is concerned, we generate $\mathcal{K}$ subsets $s_1, \cdots, s_{\mathcal{K}}$ and we insert them in S'. Each s_i is initialized to the empty set, then the vertices of G not yet inserted into a subset s_j, $j < i$, are considered sequentially according to their identifiers. A vertex is inserted into s_i if its insertion is feasible with respect to constraints (12) and (13), otherwise it is skipped. On the other hand, when the DTICP is considered, $\bar{k}$ subsets $s_1, \cdots, s_{\bar{k}}$ are generated and inserted in S'. Each s_i is initialized again to the empty set. Then the vertices of G not yet inserted into a subset s_j, $j < i$, are sequentially inserted into s_i until $|s_i| = \left\lceil \frac{|V|}{\bar{k}} \right\rceil$, or no more vertices are available. In particular, since subsets $s_1, \cdots, s_{\bar{k}}$ represent a feasible solution to the DTICP, the associated solution value is used to initialize $\bar{\bar{\theta}}$.

In any other node of the branch-and-bound tree S' includes the dummy column and the columns generated so far that are feasible with respect to the branching constraints.

Then, iteratively, the RLMP is solved to optimality, and the pricing problem is considered in order to find new negative reduced cost columns. As already mentioned in the previous section, we first try to generate columns with a negative reduced cost by means of a heuristic pricing algorithm, and if we are not successful, we solve the pricing problem using an exact MILP solver. Note that, while solving the pricing problem to optimality, all negative reduced cost columns found are returned to the master problem. If no negative reduced cost column is found, it means that the optimal solution of the current RLMP is also optimal for the LMP and the algorithm terminates.

The column generation algorithm makes also use of a restricted master heuristic. The heuristic will be described in detail in Section 4.4. The idea is to take advantage of the columns generated so far in order to obtain feasible solutions to the addressed problem. The heuristic is solved every Δ seconds, and at the end of the LMP solution process.

The availability of a primal bound helps in speeding up the convergence of the branch-and-price algorithm. For the DTICP a further consideration is needed. The cardinality of the set $S_{\bar{\bar{\theta}}}$ depends on the value of $\bar{\bar{\theta}}$. The lesser the value of $\bar{\bar{\theta}}$, the greater the dual bound generated by solving to optimality the LMP. In order to exploit the possibility to strength the dual bound generated at each node of the tree, while addressing the DTICP, we consider an improved version of the column generation algorithm. Each time a new improving feasible solution is found, the value of $\bar{\bar{\theta}}$ is updated. Also, all RLMP columns which do not satisfy constraint (21) with the updated value of $\bar{\bar{\theta}}$ are removed. Then the solution process of the current LMP is restarted.

The pseudo-code of the column generation algorithms for the DWICP and DTICP is described in Algorithm 1 and Algorithm 2, respectively. For the sake of completeness, it must be pointed out that in Algorithm 2, we keep in the pool of columns only columns that are feasible with respect to the best primal bound value $\bar{\bar{\theta}}$ found so far.

4.3 Branching scheme

Let S be equal to $S_{\bar{\theta}}$ and $S_{\bar{\bar{\theta}}}$, and let σ^* or (θ^*, σ^*) be the optimal solution of the current LMP while addressing the DWICP and the DTICP, respectively.

When the optimal solution of the LMP is fractional with respect to σ^*, we hierarchically apply the following branching rules.

If the value of $\sum_{s\in S} \sigma_s^*$ is fractional, we impose $\sum_{s\in S} \sigma_s \leq \lfloor \sum_{s\in S} \sigma_s^* \rfloor$ in one branch, and $\sum_{s\in S} \sigma_s \geq \lfloor \sum_{s\in S} \sigma_s^* \rfloor + 1$ in the other. Secondly, we apply the standard branching rules for the graph coloring problem (see [15] and [2]). In particular, given two vertices u and v, we force them to be in different subsets on one branch (which corresponds to setting $x_u + x_v \leq 1$ in the pricing problem), and to be in the same subset on the other

Algorithm 1 Column generation algorithm for the solution of the LMP for the DWICP

Step 1. RMHTime = clock(). iteration = 0.
Step 2. Initialize the RLMP with a subset $S' \subseteq S$.
Step 3. Solve the RLMP.
Step 4. $iteration \leftarrow iteration + 1$.
if $((root)$**and**$(iteration = 10))$ **or** $((clock() - RMHTime) \geq \Delta)$ **then**
  Step 5. Apply the restricted master heuristic; let UB be the value of the feasible solution found.
  Step 6. RMHTime = clock(). Go to Step 3.
end if
Step 7. Solve the pricing problem heuristically. Let P be the set of negative reduced cost columns found.
if $P \neq \emptyset$ **then**
  Step 8. $S' \leftarrow S' \cup P$. Go to Step 3.
end if
Step 9. Solve the pricing problem to optimality. Let P be the set of negative reduced cost columns found.
if $P \neq \emptyset$ **then**
  Step 10. $S' \leftarrow S' \cup P$. Go to Step 3.
end if
Step 11. Store the optimal solution of the current RLMP, which is optimal for the LMP.
Step 12. Apply the restricted master heuristic; let UB be the value of the feasible solution found.

Algorithm 2 Column generation algorithm for the solution of the LMP for the DTICP

Step 1. *RMHTime = clock(). iteration = 0.*
Step 2. Initialize the RLMP with a subset $S' \subseteq S$.
Step 3. Solve the RLMP.
Step 4. $iteration \leftarrow iteration+1$.
if $((root)$**and**$(iteration = 10))$ **or** $((clock() - RMHTime) \geq \Delta)$ **then**
  Step 5. Apply the restricted master heuristic; let UB be the value of the solution found.
  if $(UB < \bar{\bar{\theta}})$ **then**
    Step 5.1. Remove from the RLMP all infeasible columns w.r.t. UB.
    Step 5.2. Remove from the column pool all infeasible columns w.r.t. UB.
    Step 5.3. $\bar{\bar{\theta}} = UB$
  end if
  Step 6. *RMHTime = clock().* Go to Step 3.
end if
Step 7. Solve the pricing problem heuristically. Let P be the set of negative reduced cost columns found.
if $P \neq \emptyset$ **then**
  Step 8. $S' \leftarrow S' \cup P$. Go to Step 3.
end if
Step 9. Solve the pricing problem to optimality. Let P be the set of negative reduced cost columns found.
if $P \neq \emptyset$ **then**
  Step 10. $S' \leftarrow S' \cup P$. Go to Step 3.
end if
Step 11. Store the optimal solution (θ^*, σ^*) of the current RLMP, which is optimal for the LMP.
if (σ^* is integer) **then**
  Step 12. Let UB be the value of the feasible solution found.
  if $(UB < \bar{\bar{\theta}})$ **then**
    Step 12.1. Remove from the RLMP all infeasible columns w.r.t. UB.
    Step 12.2. Remove from the column pool all infeasible columns w.r.t. UB.
    Step 12.2. $\bar{\bar{\theta}} = UB$
  end if
else
  Step 13. Apply the restricted master heuristic; let UB be the value of the solution found.
  if $(UB < \bar{\bar{\theta}})$ **then**
    Step 13.1. Remove from the RLMP all infeasible columns w.r.t. UB.
    Step 13.2. Remove from the column pool all infeasible columns w.r.t. UB.
    Step 13.3. $\bar{\bar{\theta}} = UB$ and *RMHTime = clock().* Go to Step 3.
  end if
end if

branch (which corresponds to setting $x_u = x_v$ in the pricing problem). The pair u, v is chosen as follows: we compute $\mu_{uv} = \sum_{s \in S: u,v \in s} \sigma^*_s$, where σ^* is the optimal solution of the LMP; we then branch on the pair u, v with the most fractional value μ_{uv} (ties are broken at random). It is worth noting that the former branching rule applies only when the DWICP is considered.

4.4 Restricted master heuristic

In order to speed up the convergence of the branch-and-price algorithms, as shown in Section 4.2, we embed in the column generation algorithm a restricted master heuristic [14]. The basic idea behind restricted master heuristics is to solve the MP, restricted to a subset of the generated columns, by means of a general solver. The main issue to address while designing restricted master heuristics is to suitably identify the set of columns over which the MP is solved. This in order to avoid infeasibility and the device of problem dependent procedures to recover feasibility. To this aim we adopt the same technique as described in [5].

Let $\bar{S}$ denote the set of columns used in the restricted master heuristic. We first generate at most $\bar{n}_O$ subsets $C_1, \cdots, C_r$ of columns ($r \leq \bar{n}_O$), and then include all the columns contained in these subsets into $\bar{S}$. Each subset C_i is constructed as follows. Columns in the RLMP and not yet inserted in $\bigcup_{j \leq i} C_j$ are considered in non decreasing order of reduced cost. A column is added to C_i if it has at least one vertex that is not covered (colored) by the columns already contained in C_i. The inclusion of columns into C_i ends when either all vertices are covered by the columns in C_i or when there are no more columns available.

The value of $\bar{n}_O$ is dynamically adjusted after each restricted master heuristic solution as follows. If the problem is infeasible, has been solved to optimality, or the solution found is close to the corresponding dual bound, $\bar{n}_O$ is increased. Otherwise $\bar{n}_O$ is decreased. Increasing and decreasing are allowed provided that $\bar{n}_O$ stays in the interval $[\bar{n}_{min}, \bar{n}_{max}]$. A similar restricted master heuristic has been successfully applied also in [6, 3, 4].

5 Experimental results

To test the proposed branch-and-price algorithms, we have generated random instances, trying to be as close as possible to real instances of the application described in Section 1. Each instance is characterized by a number n of mobiles, a number t of antennas, and a value for parameter γ (which has an important impact on the power of the signal received at an antenna from a mobile). Each instance was obtained as follows.

- We have first generated n random points $p_1, \cdots, p_n$ according to a uniform distribution in the Euclidean plane $[100 \times 100]$. They represent the positions of n mobiles

$m_1, \cdots, m_n$, and correspond to the vertices of the considered instance.

- We have then generated t new random points $q_1, \cdots, q_t$, again using a uniform distribution in the Euclidean plane $[100 \times 100]$. They represent the positions of t antennas $b_1, \cdots, b_t$. Each mobile is assigned to the closest antenna, and we denote $s(i)$ the index of the antenna to which mobile m_i is assigned (i.e., m_i is assigned to antenna $b_{s(i)}$). In order to fit as much as possible with real instances, we forbid situations where two antennas are too close to each other. This issue is addressed as follows. In average, if the set of antennas covers all the space and each of them covers the same area, each antenna covers a surface equal to $\frac{100^2}{t}$ in the considered plane. Assuming that the area covered is approximately a circle, the radius of such a circle would be r, with $\pi r^2 = \frac{100^2}{t}$. Hence $r = \frac{100}{\sqrt{\pi t}}$, which means that the average distance between two antennas would be $2r = \frac{200}{\sqrt{\pi t}}$. But in real cases, the antennas are not perfectly distributed. We have decided that it is unrealistic to have two antennas at a distance smaller than $\frac{2r}{10} = \frac{20}{\sqrt{\pi t}}$. Thus, while generating the t points $q_1, \cdots, q_t$, we have rejected points if they were at a distance smaller than $\frac{20}{\sqrt{\pi t}}$ from already generated points.

- For each pair (i, j) with $1 \leq i \leq n$ and $1 \leq j \leq t$ we have then generated a real number $\alpha_{ij} = \frac{(X^2+Y^2)10^{Z/10}}{2}$, where X and Y are Gaussian random variables $N(0,1)$ while Z is a Gaussian random variable $N(0,8)$, and we have then set the weight W_{m_i} of each vertex m_i and the weight $\omega_{m_i m_j}$ of each arc (m_i, m_j) as follows:

$$W_{m_i} = \frac{\alpha_{is(i)}}{|q_{s(i)} - p_i|^\gamma}$$

$$\omega_{m_i m_j} = \frac{\alpha_{is(j)}}{|q_{s(j)} - p_i|^\gamma}$$

  where $|x - y|$ is the Euclidean distance between x and y.

We first analyze the proposed branch-and-price algorithm for the DWICP on instances with $t = 5$ and $n \in \{5t, 10t, 20t\}$. As already mentioned, γ represents the reduction of power density (attenuation) of an electromagnetic wave as it propagates through space. Its value is normally in the range of 2 to 4 and we have therefore decided to choose γ in $\{2, 3, 4\}$. Since $\bar{\theta} > 1$ does not physically apply, while the current technology can easily deal with $\bar{\theta} = \frac{1}{8}$, we have decided to perform tests with $\bar{\theta}$ in $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$.

The branch-and-price algorithms were implemented in C++. Both algorithms were run on a Windows 7 operating system and compiled under Visual C++ 2010 Express Edition. The experiments were carried out on an Intel Xeon processor W3680, 3.33 GHz machine with 12 GB of RAM. CPLEX 12.2.0.2 (32 bit version) was used to solve the linear relaxation of the MPs, the pricing problem and the restricted MPs of the restricted master heuristic. The overall execution time limit for each run was set to 1 hour. In particular, we allowed CPLEX to be executed in parallel on 6 cores, for an overall maximum CPU time of 6 hours.

According to a preliminary testing phase we have fixed the parameters of the branch-and-price algorithms as follows. All negative reduced cost columns found while solving the pricing problems are inserted in the RLMP. The time limit for each individual run of the restricted master heuristic is set to 1 minute. The parameter $\bar{n}_O$ is initially set to 80, while $\bar{n}_{min}$ and $\bar{n}_{max}$ are set to 15 and 150, respectively. When it is required, $\bar{n}_O$ is increased or decreased by 5. A feasible solution to the restricted MP is considered close to the corresponding dual bound if the gap is less than 5%. Finally, while addressing the DWICP and the DTICP, Δ is set equal to 900 and 300 seconds, respectively (see Algorithms 1 and 2).

Table 1 contains the results of the experiments for the DWICP. The first four columns indicate the values of t, n, γ and $\bar{\theta}$. The next two columns provide information on the root node solution: the optimal value of the LMP appears under column labeled '$\underline{z}^*(\%)$', and is expressed as a percentage of the best lower bound obtained at the end of the branch-and-price algorithm; the time (in seconds) required to compute it is reported in the next column. The last four columns give the best lower and upper bounds obtained at the end of the branch-and-price algorithm (columns '$\underline{z}^*$' and '$\overline{z}^*$', respectively), the percentage gap between these two values (column 'gap'), and, in case of a null gap, the time (in seconds) required to prove optimality (column t).

Table 2 contains exactly the same information as Table 1 but for instances with larger values of t, and Figure 3 compares the optimal number of colors for the instances with $n = 100$, $t \in \{5, 10, 15\}$, $\gamma \in \{3, 4\}$ and $\bar{\theta} \in \{\frac{1}{2}, \frac{1}{4}\}$. Such a comparison shows, for example, that for $\gamma = 3$ and $n = 100$, 44 different colors are needed if $t = 10$ and $\bar{\theta} = \frac{1}{4}$, and approximately the same number of colors (43 instead of 44) is needed when $t = 5$ and $\bar{\theta} = \frac{1}{2}$. This means that half of the 10 antennas can be avoided if the technology makes it possible to use a signal-to-interference ratio $\frac{1}{\bar{\theta}}$ of 2 instead of 4. Similarly, for $\bar{\theta} = \frac{1}{4}$ and $n = 100$, 52 different colors are needed if $t = 5$ and $\gamma = 3$, and exactly the same number of colors is needed when $t = 10$ and $\gamma = 4$. Hence, half of the 10 antennas can be avoided in an area with an attenuation factor γ of 3 instead of 4.

We can observe that for fixed values of t, n and γ, the computing time increases with $\bar{\theta}$. We also observe that the optimal value of the LMP at the root node is typically very close to the optimal value.

Table 1: DWICP for $t = 5$

t	n	γ	$\bar{\theta}$	Branch-and-price					
				root node		final solution			
				$\underline{z}^*(\%)$	t	$\underline{z}^*$	$\overline{z}^*$	gap	t
5	25	2	1	100.00	0.036	13	13	0	0.037
5	25	2	0.5	100.00	0.017	18	18	0	0.018
5	25	2	0.25	100.00	0.013	20	20	0	0.014
5	25	2	0.125	100.00	0.012	24	24	0	0.013
5	25	3	1	92.86	0.576	11	11	0	0.796
5	25	3	0.5	100.00	0.037	15	15	0	0.039
5	25	3	0.25	100.00	0.018	18	18	0	0.019
5	25	3	0.125	100.00	0.012	22	22	0	0.014
5	25	4	1	96.97	0.058	11	11	0	0.159
5	25	4	0.5	100.00	0.031	14	14	0	0.032
5	25	4	0.25	100.00	0.027	15	15	0	0.028
5	25	4	0.125	100.00	0.019	17	17	0	0.021
5	50	2	1	97.10	0.189	23	23	0	0.673
5	50	2	0.5	100.00	0.095	27	27	0	0.100
5	50	2	0.25	100.00	0.091	36	36	0	0.095
5	50	2	0.125	100.00	0.059	39	39	0	0.063
5	50	3	1	98.33	1.073	24	24	0	1.629
5	50	3	0.5	99.11	0.466	28	28	0	1.237
5	50	3	0.25	100.00	0.080	35	35	0	0.085
5	50	3	0.125	100.00	0.063	41	41	0	0.067
5	50	4	1	99.55	2.458	17	17	0	4.999
5	50	4	0.5	100.00	0.708	22	22	0	0.936
5	50	4	0.25	100.00	0.095	28	28	0	0.099
5	50	4	0.125	100.00	0.077	39	39	0	0.081
5	100	2	1	98.03	66.128	36	36	0	1597.917
5	100	2	0.5	99.44	8.237	45	45	0	12.496
5	100	2	0.25	100.00	3.292	57	57	0	7.660
5	100	2	0.125	100.00	0.624	67	67	0	0.640
5	100	3	1	96.62	134.971	34	34	0	909.294
5	100	3	0.5	97.79	6.880	43	43	0	31.699
5	100	3	0.25	99.36	1.435	52	52	0	2.855
5	100	3	0.125	99.23	0.733	65	65	0	1.763
5	100	4	1	98.28	23.166	38	38	0	166.140
5	100	4	0.5	98.98	0.936	49	49	0	3.651
5	100	4	0.25	100.00	0.717	60	60	0	0.733
5	100	4	0.125	100.00	0.593	72	72	0	0.608

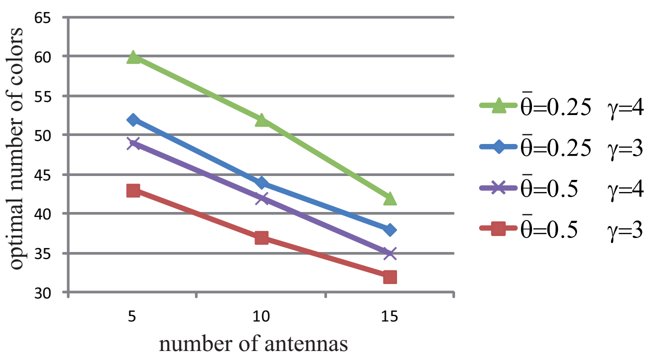


Figure 3. Comparisons of different instances with $n = 100$.

Table 2: DWICP for $t = 10$ and 15

t	n	γ	$\bar{\theta}$	Branch-and-price					
				root node		final solution			
				$\underline{z}^*$(%)	t	$\underline{z}^*$	$\overline{z}^*$	gap	t
10	50	2	0.5	100.00	0.205	23	23	0	0.210
10	50	2	0.25	100.00	0.098	29	29	0	0.103
10	50	2	0.125	100.00	0.073	37	37	0	0.077
10	50	3	0.5	96.88	0.231	24	24	0	0.748
10	50	3	0.25	100.00	0.093	30	30	0	0.098
10	50	3	0.125	100.00	0.081	34	34	0	0.086
10	50	4	0.5	100.00	1.038	18	18	0	1.225
10	50	4	0.25	100.00	0.121	24	24	0	0.126
10	50	4	0.125	100.00	0.091	30	30	0	0.095
10	100	2	0.5	98.63	7.551	39	39	0	22.183
10	100	2	0.25	99.02	1.919	51	51	0	5.023
10	100	2	0.125	100.00	0.655	60	60	0	0.671
10	100	3	0.5	99.63	9.313	37	37	0	29.983
10	100	3	0.25	100.00	2.886	44	44	0	2.901
10	100	3	0.125	98.85	1.186	58	58	0	2.980
10	100	4	0.5	100.00	0.936	42	42	0	1.778
10	100	4	0.25	100.00	1.233	52	52	0	1.248
10	100	4	0.125	100.00	0.796	59	59	0	0.811
10	200	2	0.5	99.27	2476.582	69	84	21.74	
10	200	2	0.25	99.18	82.040	85	85	0	398.924
10	200	2	0.125	99.52	8.642	104	104	0	19.936
10	200	3	0.5	98.65	1293.133	67	82	22.39	
10	200	3	0.25	98.82	135.205	83	83	0	364.682
10	200	3	0.125	99.02	13.603	102	102	0	62.556
10	200	4	0.5	99.23	393.323	63	79	25.40	
10	200	4	0.25	99.28	110.089	79	79	0	354.339
10	200	4	0.125	99.31	11.450	97	97	0	25.490
15	75	2	0.5	99.11	8.019	28	28	0	17.129
15	75	2	0.25	100.00	0.328	35	35	0	0.343
15	75	2	0.125	100.00	0.280	43	43	0	0.296
15	75	3	0.5	97.84	13.135	28	28	0	22.776
15	75	3	0.25	100.00	0.374	35	35	0	1.903
15	75	3	0.125	100.00	0.281	45	45	0	0.297
15	75	4	0.5	98.46	5.476	27	27	0	9.906
15	75	4	0.25	100.00	0.390	33	33	0	0.406
15	75	4	0.125	100.00	0.265	40	40	0	0.281
15	100	2	0.5	97.08	24.071	37	37	0	166.764
15	100	2	0.25	100.00	5.445	42	42	0	6.771
15	100	2	0.125	100.00	1.669	50	50	0	1.685
15	100	3	0.5	98.64	56.472	32	32	0	223.330
15	100	3	0.25	99.56	8.970	38	38	0	42.744
15	100	3	0.125	100.00	2.589	47	47	0	2.605
15	100	4	0.5	100.00	1.779	35	35	0	1.794
15	100	4	0.25	100.00	0.858	43	43	0	0.874
15	100	4	0.125	100.00	0.796	52	52	0	0.811
15	150	2	0.5	98.47	888.655	51	60	17.65	
15	150	2	0.25	99.25	42.261	63	63	0	145.034
15	150	2	0.125	100.00	2.823	77	77	0	2.870
15	150	3	0.5	97.81	316.743	49	55	12.24	
15	150	3	0.25	99.42	76.190	57	57	0	124.410
15	150	3	0.125	99.31	3.074	72	72	0	7.769
15	150	4	0.5	99.15	62.532	47	47	0	200.467
15	150	4	0.25	99.26	91.062	57	57	0	198.239
15	150	4	0.125	100.00	3.088	68	68	0	8.217
15	300	2	0.5	98.16	3600	17	118	594.12	
15	300	2	0.25	99.39	3600	86	137	59.30	
15	300	2	0.125	99.90	1293.101	132	151	14.39	
15	300	3	0.5	-	-	-	-	-	
15	300	3	0.25	99.49	3600	47	130	176.60	
15	300	3	0.125	99.94	1922.313	120	142	18.33	
15	300	4	0.5	-	-	-	109	-	
15	300	4	0.25	98.93	3600	77	128	66.23	
15	300	4	0.125	100.00	44.023	123	123	0	332.966

To better analyze the behavior of the branch-and-price algorithm, we report in Table 3 some performance indicators for the instances with $t = 5$ and $n = 100$. They are representative of the results obtained with other values of t and n. The first four columns of Table 3 indicate the value of the parameters t, n, γ and $\bar{\theta}$ of the considered instance. The following eight columns refer to the column generation phase. In particular they report the number of times the pricing problem has been solved ('it.') and the number of columns generated ('col.') at the root node (the first four columns) and at non-root nodes (the next four columns), respectively, while solving the pricing problem heuristically ('Heur') and exactly ('Ex.'). The following five columns show how the total computing time is distributed among the restricted linear master problem ('RLMP'), the heuristic solution of the pricing problem ('PPs - heur.'), the exact solution of the pricing problem ('PPs - ex.'), and the restricted master heuristic ('RMHs'). The values in column 'Total' are obtained by summing up the values in the four previous columns. They are smaller than 100% because of additional tasks performed by the algorithm, such as the management of the pool of columns and of the branch-and-bound tree. The last columns contain information about the feasible solutions generated by our algorithm. The first five columns refer to feasible solutions obtained by the branch-and-price algorithm independently of the restricted master heuristic (i.e., the LMP integer optimal solutions), while the last five columns refer to the restricted master heuristic. In both cases, column 'Nbr' indicates how many feasible solutions have been produced. The four following columns report data about the first and the best feasible solution found. In particular, 'value' gives the value of the solution expressed as a percentage of the final dual bound, while 'time' is the time needed to find them, expressed as a percentage of the total computing time.

We can observe that several instances are solved at the root node. When branching is performed, and thus the number of branch-and-bound nodes is higher than 1, the highest number of pricing iterations is done at the root node, and, consequently, the majority of the columns used to solve the problem is generated at the root node. This means that, to solve the linear relaxation of non-root nodes, we need to perform a small number of pricing iterations and to add a small amount of additional columns. The columns indicating the distribution of the computing time clearly show that the exact solution of the pricing problem is typically the most time consuming part of the algorithm. The last columns of Table 3 indicate that the restricted master heuristic is able to find a larger number of feasible solutions when compared to the branch-and-bound. Also, typically the first feasible solution is found in a short amount of time and is not far from the optimal solution. When the branch-and-bound is able to find a feasible solution, this solution corresponds to the best one but it typically takes a long time to find it. This means that, without the restricted master heuristic, we would not have any upper bound until approximately the end of the algorithm. It is therefore fundamental to use the restricted master heuristic to improve the efficiency of the branch-and-price approach.

In order to analyse the proposed branch-and-price algorithm for the DTICP, we have performed tests with $\bar{k} \in [k_{min} - 1, k_{max} + 1]$, where k_{min} and k_{max} are the lower and the upper bounds found when solving the corresponding DWICP problem with $\bar{\theta}$ equal to 1 and $\frac{1}{8}$, respectively. For example, for $t = 5$, $n = 100$ and $\gamma = 2$, the best lower bound value for the DWICP with $\bar{\theta} = 1$ is 36, while the best upper bound value with $\bar{\theta} = \frac{1}{8}$ is 67, and we have therefore tested the instances with $\bar{k} \in [35, 68]$.

Figure 4 shows the relation between the number $\bar{k}$ of colors and parameter θ for instances with $t = 5$, $n \in \{50, 100\}$, and $\gamma \in \{2, 3, 4\}$. It can be clearly observed that the decrease of θ is not linear with the increase of the number of colors. For example, for $t = 5$, $n = 50$ and $\gamma = 4$, 5 colors can be saved (17 instead of 22) if $\theta = 1$ instead of $\frac{1}{2}$, while 11 colors can be saved (28 instead of 39) if $\theta = \frac{1}{4}$ instead of $\frac{1}{8}$.

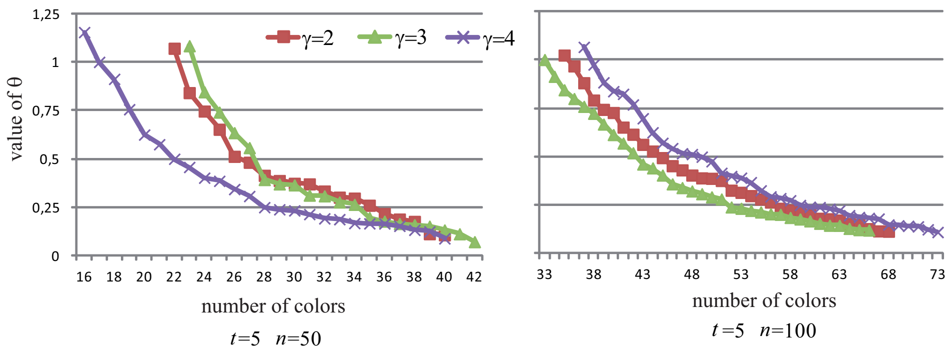


Figure 4. Relation between $\bar{k}$ and θ for instances with $t = 5$ and $n = 100$.

Instead of filling many tables with the computing times of all tested instances, we represent in Figure 5 the behavior of our branch-and-price algorithm for the DTICP on two instances, one with $t = 5$, $n = 100$, $\gamma = 2$, and the other with $t = 15$, $n = 75$ and $\gamma = 3$. The behavior of the algorithm for these instances is representative of the behavior for all remaining instances. We clearly see that the problem becomes much more difficult to solve when the number of colors is close to k_{min}. For example, for $t = 5$, $n = 100$ and $\gamma = 2$, our branch-and-price algorithm for the DTICP has found the optimal value of θ in 13 seconds for $\bar{k} = k_{max}$ while 455 seconds were necessary to determine the optimal solution for $\bar{k} = k_{min}$. For the same instance, but with $\gamma = 4$, the increase was from 13 second for k_{max} to 2491 second (i.e., more than 40 minutes) for k_{min}. As a consequence, for larger values of n, we could not solve some instances to optimality. For example, for instances with $t = 15$ and $n = 150$, we have not been able to produce any optimal solution for values of $\bar{k} \in \{k_{min} - 1, k_{min} + 6\}$.

To better analyze the behavior of the branch-and-price algorithm for the DTICP, we report in Table 4 some performance indicators for the instances with $t = 5$, $n = 100$ and for $\bar{k} \in \{k_{min}, k_1, k_2, k_{max}\}$, where k_1 and k_2 are the largest values of $\bar{k}$ with which we have been able to determine a solution of optimal value $\theta \leq \frac{1}{2}$ and $\theta \leq \frac{1}{4}$, respectively. The meaning of the columns is exactly the same as in Table 3, except that we indicate the value of $\bar{k}$ instead of $\bar{\theta}$.

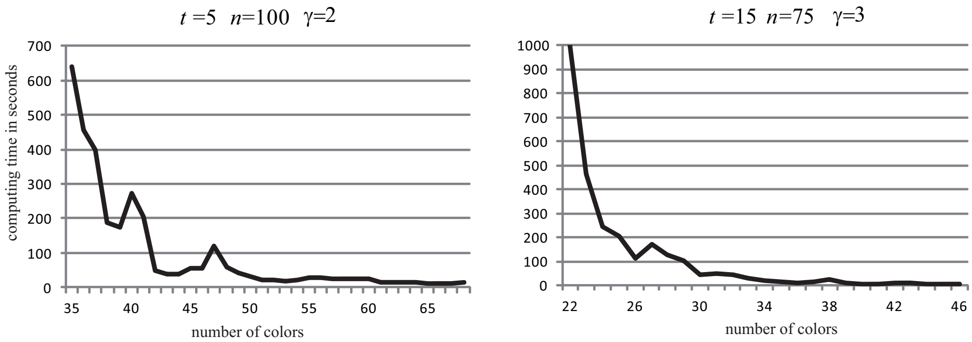


Figure 5. Computing times for the DTICP.

Here again, we can observe that many instances are solved at the root node. When branching is performed, and thus the number of branch-and-bound nodes is higher than 1, the highest number of pricing iterations is done at the root node, and, consequently, the majority of the columns used to solve the problem is generated at the root node. There is one exception in Table 4 where more than one column is generated at the non-root nodes. It is for the instance with $\gamma = 4$ and $\bar{k} = k_{min}$, where 400 columns were generated in non-root nodes (280 by the heuristic pricing algorithm and 120 by the exact solver). But even in this case, 400 is a very small number when compared to the 25788 columns generated at the root node (25774 of them by the heuristic pricing algorithm). This means that, to solve the linear relaxation of non-root nodes, we need to perform a small number of pricing iterations and to add a small amount of additional columns. The columns of Table 4 indicating the distribution of the computing time clearly show that the exact solution of the pricing problem is typically the most time consuming part of the algorithm. The last columns indicate that the branch-and-bound algorithm is not able to determine any feasible solution. In other words, the optimal solution of the LMP is always fractional. Hence, without the restricted master heuristic, we have no valid upper bound until the end of the algorithm. It is therefore fundamental to use the restricted master heuristic to improve the efficiency of the branch-and-price approach. The restricted master heuristic is able to find a few number of feasible solutions. Typically, the first feasible solution is very close to the optimal solution, but it is obtained after a large amount of time. The instance with $\gamma = 3$ and $\bar{k} = k_{min}$ is an exception where the first feasible solution was produced after only 0.08% of the total computing time. However, this feasible solution is very far from the optimal solution which corresponds to the third feasible solution produced by the restricted master heuristic after 89.37% of the total computing time.

Table 3: Performance indicators for the DWICP with $t = 5$ and $n = 100$.

t n γ	$\bar{\theta}$	Nodes	CG								Computing time in %					Feasible solutions									
			Root node				Non-root nodes				RLMP	PPs		RMHs	Total	B&B					RMHs				
			Heur.		Ex.		Heur.		Ex.			heur.	ex.			Nbr.	First		Best		Nbr.	First		Best	
			it.	col.	it.	col.	it.	col.	it.	col.							value	time	value	time		value	time	value	time
5 100 2	1	406	32	724	10	20	543	174	538	172	0.24	2.37	88.11	0.76	91.47	1	100.00	100.00	100.00	100.00	405	130.56	0.02	111.11	88.99
5 100 2	0.5	4	16	288	9	12	5	2	4	1	0.00	5.87	83.39	0.25	89.52	1	100.00	99.87	100.00	99.87	3	117.78	2.50	113.33	65.92
5 100 2	0.25	4	6	130	7	6	6	49	9	6	0.00	5.27	77.81	0.21	83.29	1	100.00	99.79	100.00	99.79	4	115.79	4.07	110.53	83.30
5 100 2	0.125	1	4	49	1	0	0	0	0	0	2.34	26.88	17.19	0.00	46.41	1	100.00	95.16	100.00	95.16	1	110.45	48.75	110.45	48.75
5 100 3	1	134	25	889	12	41	202	110	199	95	0.13	1.86	92.59	0.46	95.04	1	100.00	99.99	100.00	99.99	120	129.41	0.03	111.76	83.63
5 100 3	0.5	28	20	453	5	6	28	4	30	3	0.44	7.00	63.08	1.37	71.89	1	100.00	99.95	100.00	99.95	27	123.26	0.98	113.95	49.21
5 100 3	0.25	3	10	230	2	1	2	0	2	0	0.00	13.70	52.43	0.53	66.65	1	100.00	99.44	100.00	99.44	2	121.15	10.93	109.62	50.26
5 100 3	0.125	3	5	79	1	0	2	0	2	0	0.00	16.85	28.30	0.00	45.15	1	100.00	98.24	100.00	98.24	2	110.77	17.70	101.54	41.58
5 100 4	1	50	24	452	8	20	68	95	79	45	0.17	3.51	85.96	0.57	90.21	1	100.00	99.98	100.00	99.98	49	121.05	0.19	113.16	19.75
5 100 4	0.5	5	8	255	1	0	4	0	4	0	1.29	10.68	44.02	0.85	56.83	1	100.00	99.12	100.00	99.12	4	116.33	8.55	110.20	25.64
5 100 4	0.25	1	4	105	1	0	0	0	0	0	2.18	19.10	31.92	2.18	55.39	1	100.00	95.77	100.00	95.77	1	115.00	42.56	115.00	42.56
5 100 4	0.125	1	3	54	1	0	0	0	0	0	0.00	25.66	20.56	2.63	48.85	1	100.00	97.53	100.00	97.53	1	105.56	51.32	105.56	51.32

Table 4: Performance indicators for the DTICP with $t = 5$ and $n = 100$.

t n γ	$\bar{k}$	Nodes	CG								Computing time in %					Feasible solutions									
			Root node				Non-root nodes				RLMP	PPs		RMHs	Total	B&B					RMHs				
			Heur.		Ex.		Heur.		Ex.			heur.	ex.			Nbr	First		Best		Nbr	First		Best	
			it.	col.	it.	col.	it.	col.	it.	col.							value	time	value	time		value	time	value	time
5 100 2	36	1	27	20469	15	26	0	0	0	0	0.38	2.94	93.24	1.88	98.43	0	-	-	-	-	2	101.54	78.39	100.00	92.42
5 100 2	45	1	22	18765	7	4	0	0	0	0	2.38	19.57	59.89	5.60	87.44	0	-	-	-	-	2	102.55	78.16	100.00	91.47
5 100 2	57	1	16	9903	4	2	0	0	0	0	1.48	22.76	60.76	5.35	90.36	0	-	-	-	-	1	100.00	91.12	100.00	91.12
5 100 2	67	1	13	9704	3	1	0	0	0	0	2.23	34.48	35.66	8.24	80.60	0	-	-	-	-	1	100.00	84.82	100.00	84.82
5 100 3	34	1	29	26464	13	26	0	0	0	0	0.30	2.42	93.95	1.71	98.38	0	-	-	-	-	3	50926.11	0.08	100.00	89.37
5 100 3	43	1	22	23091	7	5	0	0	0	0	2.94	18.00	58.50	7.39	86.82	0	-	-	-	-	2	105.78	76.26	100.00	93.43
5 100 3	52	1	20	9425	4	2	0	0	0	0	2.27	24.83	60.54	4.69	92.32	0	-	-	-	-	1	100.00	91.80	100.00	91.80
5 100 3	65	1	27	7901	3	1	0	0	0	0	2.62	48.98	35.59	3.14	90.34	0	-	-	-	-	1	100.00	91.78	100.00	91.78
5 100 4	38	433	29	25774	9	14	634	280	542	120	0.50	2.80	82.18	2.99	88.47	0	-	-	-	-	214	106.27	12.57	100.00	93.98
5 100 4	49	1	15	24021	4	2	0	0	0	0	4.44	22.43	46.83	4.94	78.64	0	-	-	-	-	1	100.00	80.00	100.00	80.00
5 100 4	60	1	10	7651	3	1	0	0	0	0	3.22	23.96	49.89	7.03	84.10	0	-	-	-	-	1	100.00	86.87	100.00	86.87
5 100 4	72	1	11	7595	3	1	0	0	0	0	3.22	26.94	47.99	6.79	84.93	0	-	-	-	-	1	100.00	88.84	100.00	88.84

6 Conclusion

In this paper we have modeled a cellular channel allocation problem as a θ-improper k-coloring problem. When θ is fixed the problem, called DWICP, is to minimize the number k of colors, while when k is fixed, the problem, called DTICP, is to minimize θ. We have proposed set covering formulations for both problems, with an exponential number of variables. The models are solved through a column generation algorithm embedded in a branch-and-bound scheme, giving rise to a branch-and-price solution approach. In order to decrease the solution time, we have implemented a descent algorithm that solves the pricing problem heuristically. When variables with a negative reduced cost are generated with this heuristic, we do not solve the pricing problem to optimality. Also, we have implemented a restricted master heuristic that tries to produce feasible solutions by using some of the columns generated so far. We have observed that this heuristic is very important since it produces upper bounds which are hardly obtained by the branch-and-bound (i.e., the optimal solution of the linear relaxation of the master problem is typically fractional).

We have performed experiments on instances having the same structure as real examples of the considered cellular channel allocation problem. While instances with $n = 100$ vertices (i.e. mobiles) could be solved to optimality, we have seen that larger instances are difficult to solve, especially when the number of colors (channels) is small and parameter θ is closer to 1 than to $\frac{1}{8}$.

The proposed branch-and-price algorithms are not appropriate for real time channel allocation, since computing times should then typically be not larger than a few seconds. In such a case, heuristic algorithms must be designed for the DWICP and the DTICP, and we leave this for future developments. The branch-and-price algorithms proposed in this paper can however be used to evaluate the performance of heuristic algorithms developed in future researches.

References

[1] Araujo, J., Bermond, J.-C., Giroire, F., Havet, F., Mazauric, D., Modrzejewski, R. (2011), Weighted improper coloring, *Research report* 7590, INRIA, Sophia Antipolis.

[2] Archetti, C., Bianchessi, N., Hertz, A. (2013), A branch-and-price algorithm for the robust graph coloring problem, *Discrete Applied Mathematics*, doi: 10.1016/j.dam.2013.02.013.

[3] Archetti, C., Bianchessi, N., Hertz, A., Speranza, M.G. (2013), Incomplete service and split deliveries in a routing problem with profits. *Networks* doi: 10.1002/net.21529.

[4] Archetti, C.,Bianchessi, N., Hertz, A., Speranza, M.G. (2013), The split delivery capacitated team orienteering problem. *Networks*, doi: 10.1002/net.21519.

[5] Archetti, C., Bianchessi, N., Speranza, M.G. (2013), Optimal solutions for routing problems with profits, *Discrete Applied Mathematics* 161(4-5), 547-557.

[6] Archetti, C., Bianchessi, N., Speranza, M.G., (2011), A column generation approach for the split delivery vehicle routing problem, *Networks* 58(4), 241-254.

[7] Alouini, M.-S., Goldsmith, A.J. (1999), Area spectral efficiency of cellular mobile radio systems, *IEEE Transactions on Vehicular Technology* 48(4), 1047-1066.

[8] Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh M.W.F., Vance P..H. (1998), Branch-and-Price. Column Generation for Solving Huge Integer Programs. *Operations Research* 46(3), 316-329.

[9] Calamoneri, T. (2011), The $L(h, k)$-labeling problem: an updated survey and annotated bibliography. *The Computer Journal* 54(8),1344-1371.

[10] Cowen, L., Goddard, W., Jesurum, C.E. (1997), Defective coloring revised, *Journal of Graph Theory* 24(3), 205-219.

[11] Desaulniers, G., Desrosiers, J., Solomon, M. (2005), Column Generation, Springer.

[12] Ekstrom, H., Furuskar, A., Karlsson, J., Meyer, M., Parkvall, S., Torsner, J., Wahlqvist, M. (1999), Technical solutions for the 3G long-term evolution, *IEEE Communications Magazine* 44(3), 38-45.

[13] Haykin, S., Moher, M. (2004), Modern wireless communication, Prentice-Hall.

[14] Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., Vanderbeck, F. (2010), Column generation based primal heuristics, *Electronic Notes in Discrete Mathematics* 36, 695702.

[15] Mehrotra, A., Trick, M.A. (1996), A column generation approach for graph coloring, *INFORMS Journal on Computing* 8, 344-354.

[16] Mishra, A., Banerjee, S., Arbaugh, W. (2005), Weighted coloring based channel assignment for WLANs, *ACM SIGMOBILE Moblie Computing and Communications Review* 9(3), 19-31.

[17] Zukang Shen, Andrews, J.G., Evans, B.L. (2005), Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints, *IEEE Transactions on Wireless Communications* 4(6), 2726 - 2737.

[18] Zhang, M., Yum, T.-S.P. (1991), The nonuniform compact pattern allocation algorithm cellular mobile systems, *IEEE Transactions on Vehicular Technology* 40(2), 387-391.