

**Constructive algorithms for the
partial directed weighted improper
coloring problem**

A. Hertz, R. Montagné,
F. Gagnon

G-2015-61

June 2015

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2015.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2015.

Constructive algorithms for the partial directed weighted improper coloring problem

Alain Hertz^a

Romain Montagné^a

François Gagnon^b

^a GERAD & Department of Mathematics and Industrial
Engineering, Polytechnique Montréal, Montréal (Québec)
Canada, H3C 3A7

^b Department of Electrical Engineering, École de technologie
supérieure, Montréal (Québec) Canada, H3C 1A3

alain.hertz@gerad.ca
romain.montagne@gerad.ca
francois.gagnon@etsmtl.ca

June 2015

Les Cahiers du GERAD
G–2015–61

Copyright © 2015 GERAD

Abstract: Given a complete directed graph G 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 and an integer k , the Partial Directed Weighed Improper Coloring Problem (PDWICP) is to determine the order of the largest induced subgraph G' of G such that G' admits a θ -improper k -coloring. This problem is motivated by a practical channel assignment application where the objective is to maximize the number of simultaneously communicating mobiles in a wireless network. We consider three constructive algorithms for the standard vertex coloring problem, and adapt them to the PDWICP. We show that they perform better than today's phone operator systems based on decentralized channel assignment strategies such as fractional frequency reuse.

Key Words: Weighted improper coloring, partial coloring, channel assignment, wireless network.

1 Introduction

Let $M = \{m_1, \dots, m_n\}$ be a set of mobiles and $S = \{s_1, \dots, s_t\}$ be a set of base stations in a wireless communication network. Let P_{ip} denote the received power at s_p of a signal from m_i . We denote by $a(i)$ the index of the base station to which mobile m_i is assigned (i.e., m_i is assigned to $s_{a(i)}$), and by $c(i)$ the channel assigned to m_i . Let M_ℓ be the subset of mobiles using any given channel ℓ , i.e., $m_i \in M_\ell$ if and only if $c(i) = \ell$. An *admissible* channel assignment 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, we have the following constraints:

$$\sum_{m_j \in M_{c(i)}, j \neq i} P_{ja(i)} \leq \theta P_{ia(i)} \quad \forall i = 1, \dots, n. \quad (1)$$

where $\frac{1}{\theta}$ is the minimal admissible Signal-to-Interference Ratio (SIR). The current technology can easily deal with an SIR = 6 dB (i.e. $\frac{1}{\theta} = 8$), and sometimes with an SIR < 3 dB (i.e. $\frac{1}{\theta} < 2$).

The optimization problem we are interested in can be formulated as follows: given a set of channels and a set of mobiles assigned to their base stations, which channel should each mobile be assigned to, in order to maximize the number of mobiles respecting constraints (1)? This problem, posed by the wireless communications industry (see [1]), can be modeled as a coloring problem. More precisely, let $G = (V, A, W, \omega)$ be a directed graph with vertex set V , arc set A , and with two functions W and ω that associate a weight $W(v)$ to every vertex $v \in V$ and a weight $\omega(u, v)$ to every arc in G . Given any real number θ and any integer k , a function $c : V \rightarrow \{1, \dots, k\}$ is a θ -improper k -coloring of G if the following constraints are satisfied:

$$\sum_{(u,v) \in A \mid c(v)=c(u)} \omega(u, v) \leq \theta W(v) \quad \forall v \in V. \quad (2)$$

For a real number θ , the *Directed Weighted Improper Coloring Problem* (DWICP) is to determine the minimum integer k such that G has a θ -improper k -coloring. Since the addition of zero weight arcs does not change the problem, we can assume without loss of generality that G is complete. The DWICP was studied in [1] for the minimization of the number of channels needed to satisfy the demand of all users in a wireless network. Araujo et al. [2] have studied the DWICP 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.

The decision version of this problem, denoted k -DWICP, is to determine whether G admits a θ -improper k -coloring, where k and θ are given. The link between the above channel assignment problem and the DWICP is the following. For a set $M = \{m_1, \dots, m_n\}$ of n mobiles and a set $S = \{s_1, \dots, s_t\}$ of t base stations, we can construct a complete directed graph $G = (V, A, W, \omega)$ with vertex set $V = M$, with an arc in both directions between each pair of mobiles, and with $W(m_i) = P_{ia(i)}$ and $\omega(m_j, m_i) = P_{ja(i)}$. The construction of G is illustrated in Figure 1. 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 . We study here a slightly different problem, called *Partial Directed Weighted Improper Coloring Problem* (PDWICP). It is defined on the same directed complete graph $G = (M, A, W, \omega)$. The number k of available colors is given as well as θ , and the objective is to color as many vertices as

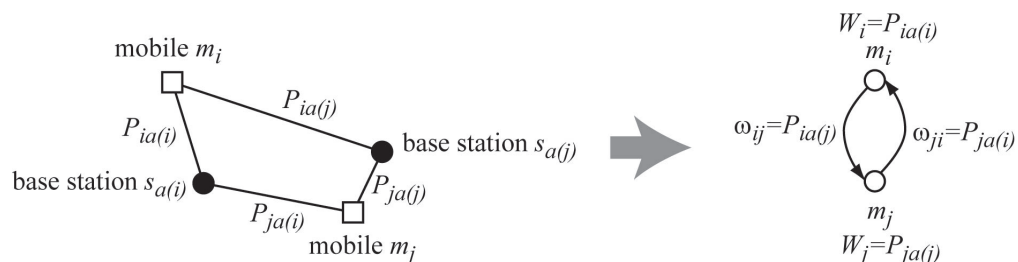


Figure 1: Construction of a directed weighted graph from a channel assignment problem.

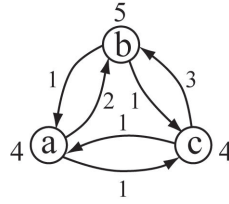


Figure 2: At most two vertices can be colored in a $\frac{1}{2}$ -improper 1-coloring of G .

possible while satisfying constraints (2). Consider for example the graph in Figure 2 with 3 vertices, where the numbers on the vertices and on the edges are their weights. If only one color is available (i.e., $k = 1$) and $\theta = \frac{1}{2}$, then only two vertices (a and b or a and c) can be colored. Indeed, if b and c receive the same color, then the sum of the weights of the arcs entering b is at least 3 which is larger than $\theta W(b) = \frac{5}{2}$.

Note that the k -DWICP is equivalent to determining whether the optimal value of the PDWICP is equal to the number $|M|$ of vertices in G . Since the k -DWICP is \mathcal{NP} -complete ([1]), we conclude that the PDWICP is \mathcal{NP} -hard.

2 Preliminaries

A region C_p called Voronoi cell is defined for each base station s_p . It contains all mobiles closer to that base station than to any other. The union of the Voronoi cells defines a Voronoi diagram [3] that divides the geographical area into t regions (where t is the number of base stations). As already mentioned, we denote by $a(i)$ the index of the base station that is the closest to mobile m_i (i.e. mobile m_i belongs to cell $C_{a(i)}$). Also, we denote by $b(i)$ the index of the second closest base station to m_i . Each cell has a center and a border defined as follows, where $\tau \in [0, 1]$ is a fixed parameter.

Definition 1 A mobile m_i is in the center of $C_{a(i)}$ if and only if $\frac{P_{ib(i)}}{P_{ia(i)}} \leq \tau$; otherwise, mobile m_i is at the border of $C_{a(i)}$.

The algorithms considered in this paper for the PDWICP are all sequential. More precisely, starting from a graph with no colored vertex (mobile), we iteratively choose an uncolored vertex and assign a color (channel) to it, until no more vertices can be colored. The assignment of a color ℓ to a mobile must be done without creating too much interference neither for mobile m_i , nor for any other mobile m_j using channel ℓ . Available channels for m_i are defined as follows, where M_ℓ denotes the set of mobiles to which channel ℓ is already assigned.

Definition 2 A channel ℓ is considered as available for $m_i \notin M_\ell$ if the following constraints are satisfied:

$$\sum_{m_j \in M_\ell} P_{ja(i)} \leq \theta P_{ia(i)}; \quad (3)$$

$$\sum_{m_h \in M_\ell, h \neq j} P_{ha(j)} + P_{ia(j)} \leq \theta P_{ja(j)} \quad \forall m_j \in M_\ell. \quad (4)$$

Constraints (3) enforce the requirement that the received power at $s_{a(i)}$ of a signal from m_i be greater, by a factor $\frac{1}{\theta}$, than the sum of interfering powers received from mobiles having channel ℓ . Constraints (4) make sure that the inclusion of m_i in M_ℓ will not create any violation of constraints (1).

The value of the solutions produced by the proposed heuristic coloring algorithms will be compared with the optimal value of the PDWICP. Such an optimal value can be obtained by solving a Boolean linear programming problem. Indeed, let x_{vi} be a binary variable that equals 1 if and only if vertex v takes color i . The PDWICP is equivalent to the following problem:

$$\max \sum_{v \in V} \sum_{i=1}^k x_{vi} \quad (5)$$

$$\text{s.t. } \sum_{i=1}^k x_{vi} \leq 1 \quad \forall v \in V \quad (6)$$

$$\sum_{u \neq v} \omega(u, v) x_{ui} \leq \theta W(v) + \alpha (1 - x_{vi}) \quad \forall v \in V, \quad \forall i \in \{1, \dots, k\} \quad (7)$$

$$x_{vi} \in \{0, 1\} \quad \forall v \in V, \quad \forall i \in \{1, \dots, k\} \quad (8)$$

The objective (5) is to maximize the number of colored vertices. Constraints (6) make sure that at most one color is assigned to every vertex. Constraints (7) are equivalent to (2), where α is a large integer (for example, $\alpha = n \max_{u,v} \omega(u, v)$) that avoids imposing any restriction if v does not have color i (i.e., if $x_{vi} = 0$).

The optimal solution of this Boolean linear program can only be obtained for relatively small instances. However, most mathematical programming solvers provide bounds on the optimal value when stopped before the end of the optimization process. We will use the CPLEX optimizer in order to obtain upper and lower bounds on the optimal solution value. Note that we could also solve the above Boolean linear model using a branch-and-price algorithm similar to the one proposed in [1], but this could be the subject of another paper. We focus here on fast heuristic algorithms that can be used for real-time channel assignment.

The rest of the paper is organized as follows. Section 3 is devoted to the description of a popular decentralized channel assignment strategy that will serve as a basis for comparison with the proposed algorithms. In Section 4, we show how to adapt three among the most popular constructive algorithms for the standard vertex coloring problem to the PDWICP. Improvements of these basic algorithms are proposed in Section 5. Computational experiments are reported in Section 6.

3 Fractional frequency reuse

Analyzing the way mobile operators currently perform channel assignment provides a lower bound for the PDWICP. Orthogonal Frequency-Division Multiple Access (OFDMA) is a multi-user version of the popular Orthogonal Frequency-Division Multiplexing (OFDM) digital modulation scheme. The latest OFDMA mobile standards such as WiMAX and LTE allocate channels using a technique called Fractional Frequency Reuse (FFR) [4].

As the name suggests, FFR implies using the same frequencies over different geographical areas. Every channel can however be used at most once in each cell, which can be written as follows, where $F = \{1, \dots, k\}$ is the set of channels:

$$|C_p \cap M_\ell| \leq 1 \quad \forall p = 1, \dots, t, \forall \ell \in F. \quad (9)$$

If a mobile is in the center (see Definition 1) of its cell C_p , then any available channel (see Definition 2) in F can be assigned to it. In order to limit co-channel interference, only a subset F_p of frequencies can be used by mobiles located at the border of C_p . The subsets F_p are typically chosen so that $|F_p| = \lfloor \frac{|F|}{3} \rfloor$ and two mobiles in the borders of two adjacent cells do not have access to the same channels. This is possible, for example, if the geographical area is divided into hexagonal cells (see Figure 3). However, when the positions of the base stations are not as regular as in Figure 3, it can happen that the number of frequencies in F_p is strictly smaller than $\lfloor \frac{|F|}{3} \rfloor$. An example is shown in Figure 4 with four base stations so that all pairs of Voronoi cells have adjacent borders. The four color theorem [5] however states that, given any separation of a plane into contiguous regions, no more than four colors are required to color the regions so that no two adjacent regions have the same color. This means that it is always possible to set $|F_p| = \lfloor \frac{|F|}{4} \rfloor$ for all cells C_p .

Taking into account the above constraints, it is not difficult to derive an upper bound on the number of users that can communicate when using the FFR technique.

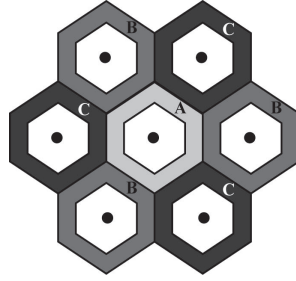


Figure 3: $F = A \cup B \cup C$; every F_p equals A , B or C ; $A \cap B = A \cap C = B \cap C = \emptyset$, $|A| = |B| = |C| = \frac{|F|}{3}$.

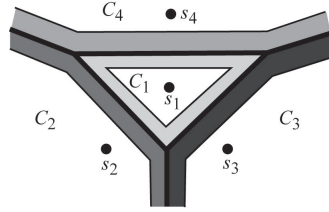


Figure 4: A base station layout where at least one F_p has at most $\frac{|F|}{4}$ frequencies.

Proposition 1 Consider a network with t base stations s_1, \dots, s_t and n mobiles m_1, \dots, m_n . Let x_p and y_p denote the number of users in the center and at the border of C_p , respectively. The following value is an upper bound on the number of simultaneously communicating mobiles when using the FFR technique:

$$\sum_{p=1}^t (\min \{x_p, |F|\} + \min \{|F_p|, y_p, \max \{0, |F| - x_p\}\}) \quad (10)$$

Proof. Consider any channel assignment that satisfies constraints (1) and (9). If no channel is available for a mobile m_i in the center of C_p while a channel ℓ is assigned to at least one mobile m_j at the border of C_p , then we can keep the number of simultaneously communicating mobiles constant by assigning channel ℓ to mobile m_i instead of m_j . As a consequence, without modifying the number of communicating mobiles, one can reach a channel assignment so that a mobile in the center of a cell C_p has no available channel only if no mobile at the border of C_p is communicating. Hence, for such an assignment, at most $\min \{x_p, |F|\}$ among the x_p users in the center of C_p are communicating. Also, at most $|F| - x_p$ channels are available for the mobiles at the border of C_p , which means that at most $\min \{|F_p|, y_p, \max \{0, |F| - x_p\}\}$ among the y_p users at the border of C_p are communicating. \square

4 Constructive algorithms for the PDWICP

The standard vertex coloring problem (VCP) is to color the vertices of an undirected graph so that no two adjacent vertices receive the same color. Many heuristics have been proposed for this problem and we show in this section how to adapt the most popular ones to the PDWICP. These algorithms are sequential and have thus relatively short execution-times, which is an important criteria for real time wireless networks.

Assume some vertices of G are already colored, and let V_ℓ ($\ell = 1, \dots, k$) denote the set of vertices having color ℓ . Suppose we want to assign color ℓ to an uncolored vertex v . In the standard VCP, such an assignment is possible only if V_ℓ does not contain any vertex adjacent to v . For the PDWICP, the coloring has to remain θ -improper when assigning color ℓ to v . In order not to violate constraints (2), we have to impose constraints similar to (3) and (4). This justifies the following definition of an *available* color for v .

Definition 3 A color ℓ is considered as available for an uncolored vertex v if the following constraints are satisfied:

$$\sum_{u \in V_\ell} \omega(u, v) \leq \theta W(v) \quad (11)$$

$$\omega(v, u) + \sum_{x \in V_\ell, x \neq u} \omega(x, u) \leq \theta W(u) \quad \forall u \in V_\ell. \quad (12)$$

Constraint (11) ensures that giving color ℓ to v will not create a violation of constraint (2) for v , while constraints (12) avoid such a violation for the vertices u already having color ℓ .

Given any partial coloring of G , we consider the following value $I_\ell(v)$ which is defined for every vertex v and every color ℓ :

$$I_\ell(v) = \sum_{u \in V_\ell, u \neq v} \omega(u, v).$$

When no vertex is colored (i.e., $V_\ell = \emptyset$ for every color ℓ), we have $I_\ell(v) = 0$ for all $v \in V$ and all $\ell \in \{1, \dots, k\}$. When assigning color ℓ to an uncolored vertex v (i.e., adding v to V_ℓ), the values $I_\ell(u)$ can be updated in $\mathcal{O}(n)$ time by setting

$$I_\ell(u) \leftarrow I_\ell(u) + \omega(v, u) \quad \forall u \in V, u \neq v.$$

Also, for an uncolored vertex v , it takes $\mathcal{O}(|V_\ell|)$ time to check whether color ℓ is available for v since it is sufficient to verify the following constraints:

$$I_\ell(v) \leq \theta W(v), \quad I_\ell(u) + \omega(v, u) \leq \theta W(u) \quad \forall u \in V_\ell.$$

For a vertex v and a set X of vertices, let $deg_X(v)$ denote the number of vertices in X adjacent to v . When selecting the next vertex v to be colored, most sequential algorithms for the VCP use decision rules based on the value $deg_X(v)$ computed with a specific set X . For example, the Welsh-Powell algorithm ([6]) chooses an uncolored vertex v with largest value $deg_V(v)$ as next vertex to be colored. For the PDWICP, we use the following measure $\mu_X(v)$, defined for all $v \in V$ and for all subset X of vertices (not necessarily including v):

$$\mu_X(v) = \frac{\sum_{u \in X, u \neq v} \omega(u, v)}{W(v)}.$$

Note that $\mu_X(v) = deg_X(v)$ when all weights (on the vertices and the edges) are equal to 1. In the sequential algorithms of the following sections, we will assume that the values $\mu_V(v)$ are computed in a pre-processing phase, and that the values $I_\ell(v)$ are available at any time (i.e., they are updated each time a vertex is colored). According to the above discussion, these initializations and updates take $\mathcal{O}(n^2)$ time and do not increase the overall complexity of any proposed algorithm since reading the input file, including the weights on all arcs, already takes $\mathcal{O}(n^2)$ time.

4.1 Adaptation of the Welsh-Powell algorithm

The Welsh-Powell algorithm ([6]) is a greedy heuristic for the VCP based on a static order of the vertices. It works as follows: the vertices are first ordered by non-increasing values $deg_V(v)$; the algorithm then considers the vertices in that order and gives to each one the smallest color (integer) not already used by at least one of its neighbors. We propose a similar algorithm for the PDWICP. More precisely, the vertices are first ordered by non increasing values $\mu_V(v)$. Each vertex v then receives the smallest available color $\ell \in \{1, \dots, k\}$ for v . If no color ℓ is available for v , then v is not colored. This algorithm, called WP1, can be implemented as shown in Table 1.

Table 1: First adaptation of the Welsh-Powell algorithm.

Algorithm 1: WP1

```

1 INITIALIZATION:
2 Determine an ordered list  $L = (v_1, \dots, v_n)$  of the vertices of  $G$  so that  $v_i$  appears earlier than  $v_j$ 
  whenever  $\mu_V(v_i) > \mu_V(v_j)$ ; set  $V_\ell = \emptyset$  for  $\ell = 1, \dots, k$ ;
3 COLORING:
4 for  $\ell = 1$  to  $k$  do
5   for  $i = 1$  to  $n$  do
6     | If  $v_i$  is not colored and color  $\ell$  is available for  $v_i$ , then add  $v_i$  to  $V_\ell$ ;
7   end
8 end

```

Proposition 2 *The WP1 algorithm runs in $\mathcal{O}(n^2)$ time.*

Proof. The ordered list L can be obtained in $\mathcal{O}(n \log n)$ time, since we assume that $\mu_V(v)$ was computed in a pre-processing phase for every vertex $v \in V$. As already mentioned, testing whether color ℓ is available for v_i requires $\mathcal{O}(|V_\ell|)$ time, which means that the final set V_ℓ is obtained in $\mathcal{O}(n|V_\ell|)$ time. Hence, the WP1 algorithm runs in $\mathcal{O}(n \sum_{\ell=1}^k |V_\ell|) \subseteq \mathcal{O}(n^2)$ time. \square

4.2 Adaptation of the Dsaturn algorithm

Given a partial solution of the VCP, the *saturation degree* of a vertex v is defined as the number of different colors appearing on vertices adjacent to v . The DSATUR algorithm, proposed by Bréaz [7], colors the vertices sequentially, choosing at each step the uncolored vertex with highest saturation degree. If several vertices maximize this value, the algorithm chooses one with a maximum number of uncolored adjacent vertices. The color assigned to the selected vertex is then the smallest color not appearing on an adjacent vertex.

In order to adapt DSATUR to the PDWICP, we define $A(v)$ as the set of available colors for an uncolored vertex v and we consider the set U of uncolored vertices v with $A(v) \neq \emptyset$. At the beginning of the algorithm, U is therefore set equal to the vertex set V , and $A(v) = \{1, \dots, k\}$ for all $v \in V$. Then, at each step of the algorithm, we consider a vertex $v \in U$ with minimum number $|A(v)|$ of available colors. In case of ties, we choose one with maximum value $\mu_U(v)$. We then remove v from U and assign to v the smallest color in $A(v)$. The sets $A(u)$ are then updated for all uncolored vertices $u \in U$ with $\ell \in A(u)$ and we remove from U the vertices u for which $A(u)$ becomes empty. The adaptation of DSATUR to the PDWICP, called DSAT1, is summarized in Table 2. Although this is not explicitly mentioned, each time a vertex is removed from U , the values $\mu_U(x)$ are updated for all vertices x remaining in U .

Table 2: First adaptation of the DSATUR algorithm.

Algorithm 2: DSAT1

```

1 INITIALIZATION:
2 Set  $U = V$  and  $V_\ell = \emptyset$  for all colors  $\ell = 1, \dots, k$ ;
3 Set  $\mu_U(v) = \mu_V(v)$ , and  $A(v) = \{1, \dots, k\}$  for all  $v \in V$ ;
4 COLORING:
5 while  $U \neq \emptyset$  do
6   | Determine a vertex  $v \in U$  with minimum value  $|A(v)|$ . In case of ties, choose one with maximum
  | value  $\mu_U(v)$ ;
7   | Choose the smallest color  $\ell \in A(v)$  and move  $v$  to  $V_\ell$ ;
8   | Update the sets  $A(u)$  for all  $u \in U$  with  $\ell \in A(u)$ , and remove from  $U$  all vertices  $u$  for which
  |  $A(u) = \emptyset$ ;
9 end

```

Proposition 3 *The DSAT1 algorithm runs in $\mathcal{O}(n^3)$ time.*

Proof. Since we assume that $\mu_V(v)$ is computed in a pre-processing phase for all $v \in V$, the initialization phase takes $\mathcal{O}(nk)$ time. Each time a vertex w is removed from U , we can update $\mu_U(u)$ for all $u \in U$ by subtracting $\frac{\omega(w,u)}{W(u)}$. Hence, the total time needed by DSAT1 to perform these updates is in $\mathcal{O}(n^2)$. Steps 6 and 7 clearly require $\mathcal{O}(n+k)$ time. When assigning color ℓ to v , we have to test whether color ℓ remains available for the uncolored vertices with $\ell \in A(u)$ (Step 8). Each such test requires $\mathcal{O}(|V_\ell|)$ operations, which means that the sets $A(u)$ can be updated in $\mathcal{O}(n|V_\ell|) \subseteq \mathcal{O}(n^2)$ time. Since there are n vertices to be colored, DSAT1 runs in $\mathcal{O}(n^3)$ time. \square

4.3 Adaptation of the RLF algorithm

The Recursive Largest First algorithm (RLF), proposed by Leighton [8] for the VCP, builds a sequence V_1, \dots, V_k of color classes. Consider the construction of one color class V_ℓ , let U denote the set of uncolored vertices, and let W be the set (initially empty) of uncolored vertices with at least one neighbor in V_ℓ . Every time a vertex in U is chosen to be moved to V_ℓ , all its neighbors in U are moved from U to W . The first vertex $v \in U$ to be included in V_ℓ is the one with the largest number $\text{deg}_U(v)$ of adjacent vertices in U . The rest of V_ℓ is built as follows : while U is not empty, the next vertex to be moved from U to V_ℓ is one having the largest number $\text{deg}_W(v)$ of neighbors in W . Ties are, if possible, broken by choosing a vertex with the smallest number $\text{deg}_U(v)$ of neighbors in U .

Table 3: First adaptation of the RLF algorithm.

Algorithm 3: RLF1

```

1 INITIALIZATION:
2 Set  $V_\ell = \emptyset$  for  $\ell = 1, \dots, k$ ;
3 COLORING:
4 for  $\ell = 1$  to  $k$  do
5   Let  $U$  be the set of uncolored vertices; compute  $\mu_U(u)$  for all  $u \in U$ . Set  $W = \emptyset$  and  $\mu_W(u) = 0$  for
   all  $u \in U$ ;
6   Choose a vertex  $v \in U$  with largest value  $\mu_U(v)$ ;
7   Move  $v$  to  $V_\ell$  and move to  $W$  all vertices in  $U$  for which color  $\ell$  is no more available;
8   while  $U \neq \emptyset$  do
9     Choose a vertex  $u \in U$  with largest value  $\mu_W(u)$ . In case of ties, choose one with smallest value
      $\mu_U(u)$ ;
10    Move  $u$  to  $V_\ell$ , and move to  $W$  all vertices in  $U$  for which color  $\ell$  is no more available;
11  end
12 end

```

The adaptation of the RLF algorithm to the DWICP is done as follows. When constructing V_ℓ , W is defined as the set of uncolored vertices v for which color ℓ is not available (i.e., at least one of constraints (11) and (12) is violated if color ℓ is assigned to v). Moreover, we use $\mu_U(v)$ and $\mu_W(v)$ instead of $\text{deg}_U(v)$ and $\text{deg}_W(v)$. The adaptation of RLF to the PDWICP, called RLF1, is summarized in Table 3. Although this is not explicitly mentioned, each time a vertex is moved from U to V_ℓ or to W , the values $\mu_U(x)$ and $\mu_W(x)$ are updated for all vertices x remaining in U .

Proposition 4 *This RLF1 algorithm runs in $\mathcal{O}(n^3)$ time.*

Proof. Consider the construction of a color class V_ℓ . The initial values $\mu_U(u)$ for all $u \in U$ can easily be obtained in $\mathcal{O}(n^2)$ time, while the rest of Steps 5 and 6 can clearly be performed in $\mathcal{O}(n)$ time. Step 7 requires $\mathcal{O}(n)$ time since a vertex u is moved from U to W if and only if $\omega(u, v) > \theta W(v)$ or $\omega(v, u) > \theta W(u)$ (where v is the vertex chosen at Step 6).

We now analyze the time needed to update $\mu_U(u)$ and $\mu_W(u)$ for $u \in U$ when vertices are moved from U to W or to V_ℓ . Each time a vertex w is moved from U to W , $\mu_W(x)$ is incremented by $\frac{\omega(w,x)}{W(x)}$ and $\mu_U(x)$ is decreased by $\frac{\omega(w,x)}{W(x)}$ for all $x \in U$. Also, when a vertex $v \in U$ is moved from U to V_ℓ , $\mu_U(x)$ is decreased by $\frac{\omega(v,x)}{W(x)}$ for all $x \in U$. Since all vertices of U are moved either to W or to V_ℓ until U becomes empty, there are $\mathcal{O}(n^2)$ such updates, each being performed in constant time.

When assigning color ℓ to a vertex, we have to check in Step 10 whether color ℓ remains available for the vertices in U . As already mentioned, this can be done in $\mathcal{O}(|U||V_\ell|) \subseteq \mathcal{O}(n|V_\ell|)$. Hence, the total time needed to perform these tests when constructing V_ℓ is in $\mathcal{O}(n|V_\ell|^2)$.

In summary, the construction of V_ℓ is done in $\mathcal{O}(n^2 + n|V_\ell|^2)$, which means that RLF1 runs in $\mathcal{O}(kn^2 + n \sum_{\ell=1}^k |V_\ell|^2) \subseteq \mathcal{O}(n^3)$ time. \square

5 Improved algorithms

Experiments reported in Section 6 will show that WP1, DSAT1 and RLF1 often assign the same color to two vertices u and v for which $\omega(u, v)$ or $\omega(v, u)$ has a large value when compared to $\theta W(v)$ or to $\theta W(u)$, respectively. This creates high interference at u or v , which implies that very few other vertices are then added to the color class shared by u and v . We offer two ways to avoid this problem.

5.1 Improvement based on the Voronoi diagram

The first improvement technique assumes that G is built from a given set $S = \{s_1, \dots, s_t\}$ of base stations and a given set $M = \{m_1, \dots, m_n\}$ of mobiles, as explained in Section 1. This implies that every vertex in G has a geographical location. As mentioned in Section 3, the FFR technique is based on a partition of the geographical area into cells. Users at the center of a cell C_p have access to the whole frequency band F , while only a subset F_p of the frequencies can be used by mobiles located at the border of C_p . We propose to partially imitate such a strategy by considering the frequencies in F_p as preferred channels for mobiles at the center of C_p . This is now described in more detail.

The Voronoi cell C_p associated with every base station s_p contains all mobiles closer to s_p than to any other base station. The dual of the Voronoi diagram is a graph $H = (V_H, E_H)$ with vertex set $V_H = S$ (i.e., every base station is a vertex), and edge set $E_H = \{(s_p, s_q) | C_p \text{ and } C_q \text{ are adjacent Voronoi cells}\}$. Let d_{pq} denote the Euclidian distance between s_p and s_q . We first color the vertices of H with as few colors as possible, trying to maximize the minimum distance between two vertices with the same color. This can be achieved by solving a Boolean linear program. Indeed, let $D = \max_{1 \leq p, q \leq t} d_{pq}$. Also let x_{pi} be a binary variable that equals 1 if and only if vertex s_p receives color i , let y_{pq} be a binary variable that equals 1 if and only if vertices s_p and s_q share the same color, and let z_i be a binary variable that equals 1 if and only if color i is used. We consider the following problem.

$$\min \quad D \sum_{i=1}^4 z_i - \delta \quad (13)$$

$$\text{s.t.} \quad \sum_{i=1}^4 x_{pi} = 1 \quad \forall s_p \in V_H \quad (14)$$

$$x_{pi} + x_{qi} \leq z_i \quad \forall (s_p, s_q) \in E_H, \quad \forall i = 1, \dots, 4 \quad (15)$$

$$x_{pi} + x_{qi} - 1 \leq y_{pq} \quad \forall s_p, s_q \in V_H, p \neq q, \quad \forall i = 1, \dots, 4 \quad (16)$$

$$\delta \leq d_{pq} + D(1 - y_{pq}) \quad \forall (s_p, s_q) \notin E_H \quad (17)$$

$$x_{pi}, y_{pq}, z_i \in \{0, 1\} \quad \forall s_p, s_q \in V_H, \quad \forall i = 1, \dots, 4 \quad (18)$$

As already mentioned in Section 3, the four color theorem [5] states that no more than four colors are required to color the vertices of H . Constraints (14) enforce the use of exactly one of the four colors for every

vertex of H . Constraints (15) express the fact that adjacent vertices in H cannot have the same color, and relate the x variables to the z variables. Constraints (16) relate the x variables to the y variables. Constraints (17) enforce δ to be at most as large as the minimum distance between two vertices with the same color. The objective (13) is to use as few colors as possible, and among such colorings, to find one with maximum value δ .

An optimal solution of this linear program is possibly difficult to obtain for large instances since it contains binary variables. The number of base stations is however typically not too large which makes it possible to obtain an optimal solution in a reasonable computing time. If this not the case, it is then possible to simply build a 4-coloring of H , which is equivalent to finding a feasible solution to the above linear program. The Four Color Theorem gives an $O(n^2)$ algorithm to determine such a coloring of the vertices of the planar graph H [9].

So, assume we have found a coloring of the vertices of H , whichever method was used to obtain such a coloring. Let $\chi \leq 4$ be the number of colors used, and let $c(p) \in \{1, \dots, \chi\}$ denote the color assigned to vertex s_p . Assume the number k of available colors is a multiple of 12. To each region C_p , we associate a subset F_p of colors so that $F_p \cap F_q = \emptyset$ if C_p is adjacent to C_q , and all sets F_p contain $\frac{k}{\chi}$ colors. This is done as follows. For every $p = 1, \dots, t$, we define $F_p = \{\frac{(c(p)-1)k}{\chi} + 1, \dots, \frac{c(p)k}{\chi}\}$. For example, if $k = 120$ channels are available, $\chi = 3$ and $c(p) = 2$, then $F_p = \{41, \dots, 80\}$, while if $k = 120$, $\chi = 4$ and $c(p) = 2$, then $F_p = \{31, \dots, 60\}$. These sets are used to define priority rules. More precisely, for every mobile m_i , we define the following set $S(m_i)$ of preferred channels for m_i : if m_i is at the border of C_p , we set $S(m_i) = F_p$, while if m_i is in the center of C_p , we set $S(m_i) = \{1, \dots, k\}$.

Remember that according to Definition 1, a mobile m_i is in the center of its cell if and only if $\frac{P_{ib(i)}}{P_{ia(i)}} \leq \tau$, where τ is a given parameter. Hence, the sets $S(m_i)$ of preferred channels depend on τ . For example, if $\tau = 0$, then all mobiles are at the border of their cell and thus all have a preference for a subset of $\frac{k}{\chi}$ channels; if $\tau = 1$, all mobiles are in the center of their cells, and no priority rule is applied. We will run our algorithms with 11 different values for τ , namely 0, 0.1, 0.2, \dots , 1, and keep the best solution.

For the rest of this section, we assume that each vertex v of G has a set $S(v)$ of preferred colors. WP2, DSAT2 and RLF2 are modified versions of WP1, DSAT1 and RLF1, that take into account these preferences by first assigning a preferred color $\ell \in S(v)$ to every vertex v , and then completing the coloring by using non-preferred colors. The three algorithms appear in Tables 4, 5 and 6. It is not difficult to observe that WP2, DSAT2, and RLF2 have the same time complexity as WP1, DSAT1 and RLF1, respectively.

Table 4: Second adaptation of the Welsh-Powell algorithm.

Algorithm 4: WP2

```

1 INITIALIZATION:
2 Determine an ordered list  $L = (v_1, \dots, v_n)$  of the vertices of  $G$  so that  $v_i$  appears earlier than  $v_j$ 
  whenever  $\mu_V(v_i) > \mu_V(v_j)$ ; set  $V_\ell = \emptyset$  for  $\ell = 1, \dots, k$ ;
3 Apply the following COLORING procedure using  $\tau = 0, 0.1, 0.2, \dots, 1$  for defining the sets of preferred
  colors, and keep the best result;
4 Procedure COLORING:
5 for  $\ell = 1$  to  $k$  do
6   for  $i = 1$  to  $n$  do
7     | If  $v_i$  is not already colored,  $\ell \in S(v_i)$ , and  $\ell$  is available for  $v_i$ , then add  $v_i$  to  $V_\ell$ ;
8   end
9   for  $i = 1$  to  $n$  do
10    | If  $v_i$  is not already colored and  $\ell$  is available for  $v_i$ , then add  $v_i$  to  $V_\ell$ ;
11  end
12 end

```

Table 5: Second adaptation of the DSATUR algorithm.

Algorithm 5: DSAT2

```

1 INITIALIZATION:
2 Set  $U = V$  and  $V_\ell = \emptyset$  for all colors  $\ell = 1, \dots, k$ ;
3 Set  $\mu_U(v) = \mu_V(v)$ , and  $A(v) = \{1, \dots, k\}$  for all vertices  $v$ ;
4 Apply the following COLORING procedure using  $\tau = 0, 0.1, 0.2, \dots, 1$  for defining the sets of preferred colors, and keep the best result;
5 Procedure COLORING:
6 while  $U \neq \emptyset$  do
7   Determine a vertex  $v \in U$  with minimum value  $|A(v)|$ . In case of ties, choose one with maximum value  $\mu_U(v)$ ;
8   If  $A(v) \cap S(v)$  contains at least one color, then choose the smallest one  $\ell$ ; otherwise, choose the smallest color  $\ell \in A(v)$ ;
9   Move  $v$  to  $V_\ell$ , update the sets  $A(u)$  for all  $u \in U$  with  $\ell \in A(u)$ , and remove from  $U$  all vertices  $u$  for which  $A(u) = \emptyset$ ;
10 end

```

Table 6: Second adaptation of the RLF algorithm.

Algorithm 6: RLF2

```

1 INITIALIZATION:
2 Set  $V_\ell = \emptyset$  for  $\ell = 1, \dots, k$ ;
3 Apply the following COLORING procedure using  $\tau = 0, 0.1, 0.2, \dots, 1$  for defining the sets of preferred colors, and keep the best result;
4 Procedure COLORING:
5 for  $\ell = 1$  to  $k$  do
6   Let  $U$  be the set of uncolored vertices, and let  $U' \subseteq U$  be the subset of uncolored vertices  $v$  with  $\ell \in S(v)$ ; compute  $\mu_U(u)$  for all  $u \in U$  and  $\mu_{U'}(u)$  for all  $u \in U'$ . Set  $W = \emptyset$  and  $\mu_W(u) = 0$  for all  $u \in U$ ;
7   If  $U' \neq \emptyset$ , choose a vertex  $v \in U'$  with largest value  $\mu_{U'}(v)$ ; otherwise choose a vertex  $v \in U$  with largest value  $\mu_U(v)$ ;
8   Move  $v$  to  $V_\ell$  and move to  $W$  all vertices in  $U$  (hence also those in  $U'$ ) for which color  $\ell$  is no more available;
9   while  $U' \neq \emptyset$  do
10    Choose a vertex  $u \in U'$  with largest value  $\mu_W(u)$ . In case of ties, choose one with smallest value  $\mu_{U'}(u)$ ;
11    Move  $u$  from  $U'$  to  $V_\ell$ , and move to  $W$  all vertices in  $U$  for which color  $\ell$  is no more available;
12  end
13  while  $U \neq \emptyset$  do
14    Choose a vertex  $u \in U$  with largest value  $\mu_W(u)$ . In case of ties, choose one with smallest value  $\mu_U(u)$ ;
15    Move  $u$  from  $U$  to  $V_\ell$ , and move to  $W$  all vertices in  $U$  for which color  $\ell$  is no more available;
16  end
17 end

```

5.2 Super-available colors

The use of preferred colors proposed in the previous section is not always sufficient to avoid assigning the same color to two vertices u and v for which $\omega(u, v)$ or $\omega(v, u)$ has a large value when compared to $\theta W(v)$ or $\theta W(u)$, respectively. We present here another way to avoid this problem. It is based on the following definition that depends on a real parameter $\rho \in [0, 1]$.

Definition 4 A color ℓ is super-available for a vertex v if in addition to constraints (11), (12), the following ones are also satisfied:

$$\omega(u, v) \leq \rho(\theta W(v)) \quad \text{and} \quad \omega(v, u) \leq \rho(\theta W(u)) \quad \forall u \in V_\ell.$$

Algorithms WP3, DSAT3, and RLF3 described here below first try to color the vertices using only super-available colors. The coloring process is then repeated with the uncolored vertices, by considering all available colors. It is not difficult to observe that the proposed changes do not affect the algorithms' complexity. We will run our algorithms with 11 different values for ρ , namely 0, 0.1, 0.2, \dots , 1, and keep the best solution.

Table 7: Third adaptation of the Welsh-Powell algorithm.

Algorithm 7: WP3

- 1 **INITIALIZATION:**
- 2 Determine an ordered list $L = (v_1, \dots, v_n)$ of the vertices of G so that v_i appears earlier than v_j whenever $\mu_V(v_i) > \mu_V(v_j)$; Set $V_\ell = \emptyset$ for $\ell = 1, \dots, k$;
- 3 Apply the following COLORING procedure using $\rho = 0, 0.1, 0.2, \dots, 1$ for the definition of super-available colors, and keep the best result;
- 4 Procedure **COLORING:**
- 5 **for** $\ell = 1$ to k **do**
- 6 **for** $i = 1$ to n **do**
- 7 If v_i is not already colored and color ℓ is super-available for v_i , then add v_i to V_ℓ ;
- 8 **end**
- 9 **for** $i = 1$ to n **do**
- 10 If v_i is not already colored and color ℓ is available for v_i , then add v_i to V_ℓ ;
- 11 **end**
- 12 **end**

Table 8: Third adaptation of the DSATUR algorithm.

Algorithm 8: DSAT3

- 1 **INITIALIZATION:**
- 2 Set $U = V$ and $V_\ell = \emptyset$ for all colors $\ell = 1, \dots, k$;
- 3 Set $\mu_U(v) = \mu_V(v)$, and $A(v) = \{1, \dots, k\}$ for all vertices v ;
- 4 Apply the following COLORING procedure using $\rho = 0, 0.1, 0.2, \dots, 1$ for the definition of super-available colors, and keep the best result;
- 5 Procedure **COLORING:**
- 6 **while** $U \neq \emptyset$ **do**
- 7 Determine a vertex $v \in U$ with minimum value $|A(v)|$. In case of ties, choose one with maximum value $\mu_U(v)$;
- 8 If $A(v)$ contains at least one super-available color for v , choose the smallest one ℓ and move v to V_ℓ ; otherwise, choose the smallest color $\ell \in A(v)$ and move v to V_ℓ ;
- 9 Update the sets $A(u)$ for all $u \in U$ with $\ell \in A(u)$, and remove from U all vertices u for which $A(u) = \emptyset$;
- 10 **end**

6 Numerical results

In this section, we compare the optimal value provided by the Boolean linear programming model of Section 2 with the values produced by the proposed heuristics. Such a comparison is only possible for instances of relatively small size. For larger instances, we compare the results of our heuristics with an upper bound on the performance of the Fractional Frequency Reuse (FFR) technique.

Table 9: Third adaptation of the RLF algorithm.

Algorithm 9: RLF3

```

1 INITIALIZATION:
2 Set  $V_\ell = \emptyset$  for  $\ell = 1, \dots, k$ ;
3 Apply the following COLORING procedure using  $\rho = 0, 0.1, 0.2, \dots, 1$  for the definition of
  super-available colors, and keep the best result;
4 Procedure COLORING:
5 for  $\ell = 1$  to  $k$  do
6   Let  $U$  be the set of uncolored vertices; let  $U_1 \subseteq U$  be the subset of uncolored vertices  $v$  for which  $\ell$ 
   is super-available, and let  $U_2 = U \setminus U_1$ ; compute  $\mu_U(u)$  for all  $u \in U$  and  $\mu_{U_1}(u)$  for all  $u \in U_1$ . Set
    $W = \emptyset$  and  $\mu_W(u) = 0$  for all  $u \in U$ ;
7   If  $U_1 \neq \emptyset$ , choose a vertex  $v \in U_1$  with largest value  $\mu_{U_1}(v)$ ; otherwise, choose a vertex  $v \in U$  with
   largest value  $\mu_U(v)$ ;
8   Move  $v$  to  $V_\ell$ ; move to  $W$  all vertices in  $U$  for which color  $\ell$  is no more available; move to  $U_2$  all
   vertices in  $U_1$  for which color  $\ell$  is no more super-available;
9   while  $U_1 \neq \emptyset$  do
10    Choose a vertex  $u \in U_1$  with largest value  $\mu_W(u)$ . In case of ties, choose one with smallest
    value  $\mu_{U_1}(u)$ ;
11    Move  $u$  to  $V_\ell$ ; move to  $W$  all vertices in  $U$  for which color  $\ell$  is no more available; move to  $U_2$  all
    vertices in  $U_1$  for which color  $\ell$  is no more super-available;
12  end
13  while  $U \neq \emptyset$  do
14    Choose a vertex  $u \in U$  with largest value  $\mu_W(u)$ . In case of ties, choose one with smallest value
     $\mu_U(u)$ ;
15    Move  $u$  to  $V_\ell$ ; move to  $W$  all vertices in  $U$  for which color  $\ell$  is no more available;
16  end
17 end

```

All algorithms are tested on random instances generated in the same way as in [1]. The positions of the mobiles and of the base stations are randomly generated according to a uniform distribution in the Euclidian plane, forbidding situations where base stations are too close to one another. The received power P_{ip} at s_p of a signal from mobile 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 is the product of an exponential random variable to denote short term fading and a log-normal random variable for shadowing,
- γ is the attenuation factor which typically varies from 2 in free-space to 4 for Non-Line Of Sight (NLOS) in urban areas.

For more details about how these instances were generated, the reader is invited to consult [1].

6.1 Comparisons with optimal solutions

We denote by OPT the optimal value provided by the Boolean linear programming model of Section 2, and compare it with the values produced by the nine proposed heuristics WP1, WP2, WP3, DSAT1, DSAT2, DSAT3, RLF1, RLF2 and RLF3. Since OPT can only be computed for instances of relatively small size, we have limited these comparisons to instances with $t = 10$ base stations, $n = 25, 30, 35, 40$ mobiles, $k = 12$ channels and $(\theta, \gamma) = (0.25, 4)$. For each combination of parameters t, n, k, θ, γ , we have generated 25

instances, for a total of 100 instances. Average results are shown in Figure 5. In abscissa, we consider various maximum allowed gaps (expressed in percentage) between OPT and the value produced by the heuristics, while the ordinate indicates the percentage of instances for which the given gap was not exceeded. For example, considering WP1, we can read that 88% of the instances were solved to optimality, while the maximum gap between OPT and the obtained value has never exceeded 6% for 99% of the instances, and 96% of the instances were solved with a maximum gap of 4%.

It clearly appears that WP2 performs better than WP1, and WP3 outperforms WP1 and WP2, producing 94% optimal solutions. The maximum gap between OPT and the WP3 value has never exceeded 6%. The analysis is similar for the other heuristics: DSAT3 outperforms DSAT1 and DSAT2 with a total of 95% optimal solutions and a maximum gap of 3%. Finally, RLF3 produces 94% optimal solutions in comparison with 90% for RLF1 and 95% for RLF2. The maximum gap between OPT and the RLF3 value is always below 6%.

In the three other performance profiles of Figure 5, WP_{*i*}, DSAT_{*i*} and RLF_{*i*}, (*i* = 1, 2, 3) are compared with each other. It turns out that DSAT1 performs better than WP1 and RLF1, while RLF2 is better than

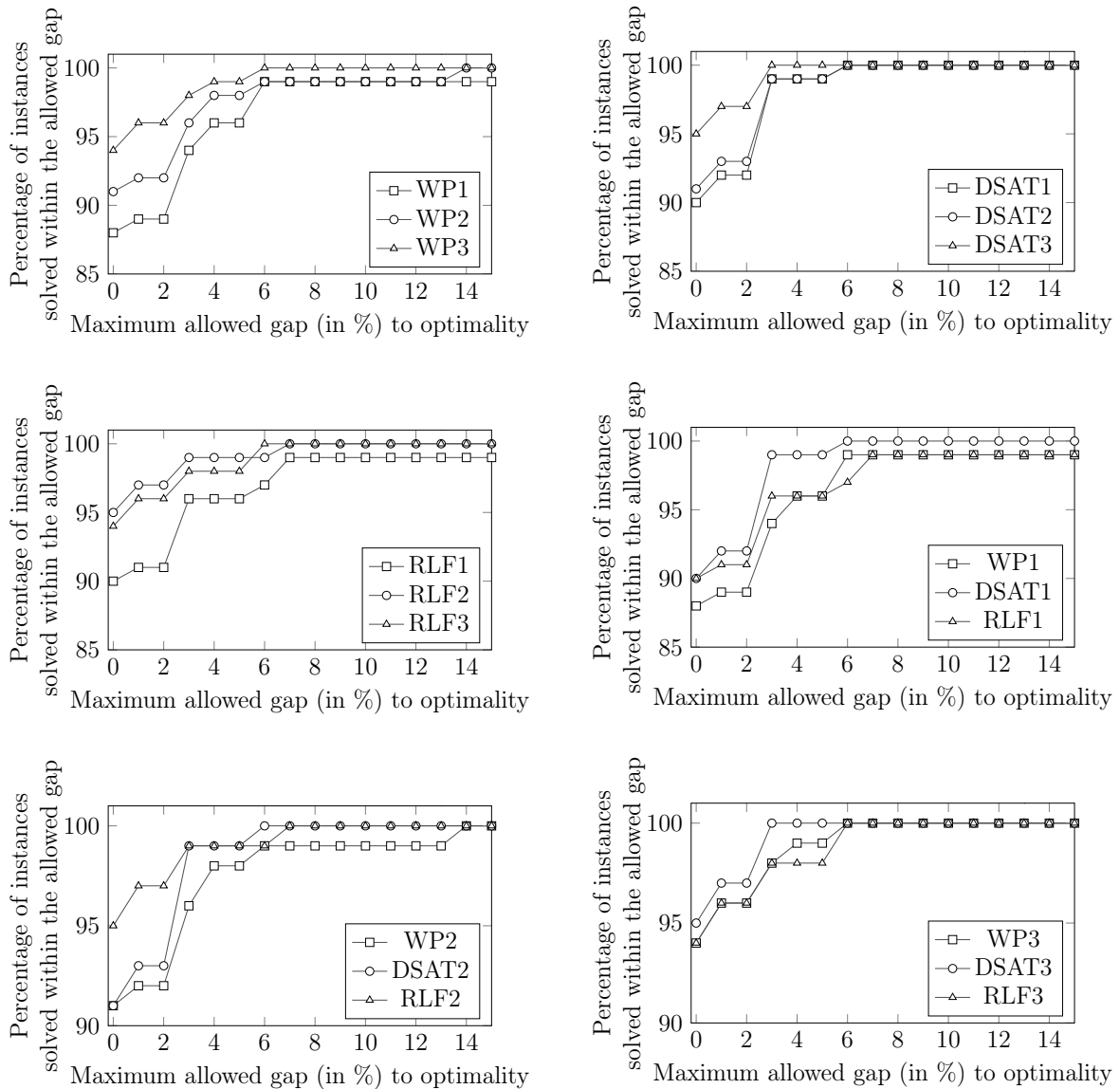


Figure 5: Comparisons between exact and heuristic algorithms.

DSAT2 and WP2, and DSAT3 is better than WP3 and RLF3. We also notice that the maximum gap between OPT and the DSAT i value is always smaller than the same gap computed with WP i or RLF i : it is equal to 6% for DSAT1 and DSAT2, and to only 3% for DSAT3.

These performance profiles clearly show that the proposed heuristics are able to reach optimality for a large number of small instances. Indeed, all heuristics, except WP1, have produced at least 90% optimal solutions. Also, the proposed improvements of Section 5 are clearly effective since versions 2 and 3 of the proposed heuristics perform better than version 1.

While DSAT3 could appear as the winner of this comparison, it is important to mention that WP3 is also attractive as it runs in $\mathcal{O}(n^2)$ time while DSAT3 and RLF3 run in $\mathcal{O}(n^3)$ time. This can clearly be observed in Table 10 where we report computing times (in seconds) for realistic networks having $n = 500$ to 1100 mobiles, $k = 120$ channels, $t = 25$ base stations, and $(\theta, \gamma) = (0.25, 4)$. The tests were run on a 3.4 GHz Intel(R) Core(TM) i7-2600 CPU machine with 16 GB of RAM.

Table 10: Average CPU time (in seconds) with $(\theta, \gamma) = (0.25, 4)$, $t = 25$ base stations, $k = 120$ channels.

n	WP1	WP2	WP3	DSAT1	DSAT2	DSAT3	RLF1	RLF2	RLF3
500	0.20	0.22	0.30	0.78	0.83	0.85	7.10	7.10	8.22
550	0.30	0.30	0.40	0.98	1.00	1.03	9.40	9.40	11.52
600	0.33	0.42	0.48	1.28	1.32	1.37	13.65	13.65	16.28
650	0.37	0.43	0.58	1.52	1.50	1.62	16.28	16.28	19.68
700	0.42	0.43	0.60	1.53	1.65	1.62	20.05	20.05	22.30
750	0.50	0.57	0.68	1.88	1.80	1.87	23.15	23.15	28.45
800	0.58	0.67	0.83	2.27	2.32	2.48	32.23	32.23	34.93
850	0.70	0.75	0.95	2.68	2.60	2.72	38.33	38.33	36.63
900	0.73	0.73	0.92	2.67	2.62	2.90	40.87	40.87	38.63
950	0.77	0.82	1.02	2.63	2.50	2.78	55.35	55.35	48.33
1000	0.90	0.87	1.03	3.05	3.13	3.37	59.75	59.75	45.78
1050	0.95	1.00	1.20	3.67	3.58	3.97	66.20	66.20	45.22
1100	1.10	1.12	1.32	3.97	4.03	4.58	79.75	79.75	53.63

Note that even if versions 2 and 3 of our algorithms run the COLORING procedure 11 times, the total time needed for these 11 runs is comparable with one run of version 1. The reason is that the selection of the color to assign to a vertex is much faster in versions 2 and 3 since we limit our choices to preferred or super-available colors.

6.2 Parameter adjustments

Versions 2 and 3 of the proposed heuristics depend on parameters $\tau \in [0, 1]$ and $\rho \in [0, 1]$, respectively. We have performed tests to see if specific values of these parameters should be preferred to others. As in the previous section, we have considered instances with $t = 10$, $n = 25, 30, 35, 40$, $k = 12$, $\theta = 0.25$ and $\gamma = 4$, and we have generated 25 instances for each combination of t, n, k, θ, γ , for a total of 100 instances. Average results produced by WP2 and WP3 are given in Figures 6 and 7, for τ and ρ equal to 0, 0.25, 0.5, 0.75 and 1. Hence, instead of reporting the best among 11 runs of the COLORING procedure, with different values of τ or ρ (as was the case in Figure 5), we report here the individual results produced by the COLORING procedure with each of the 5 considered values of τ or ρ .

We observe that about 80% optimal solution are obtained with $\tau \geq 0.75$ while 73% of them are found with $\tau = 0.5$, and about 70% with $\tau \leq 0.25$. However, additional tests on larger instances have shown that there seems to be no general rule for choosing a good value for τ . For example, these additional tests have shown that $\tau = 0.75$ should be preferred to $\tau = 0.5$ 27% of the time, while preference should be given to $\tau = 0.5$ over $\tau = 0.75$ 13% of the time. This explains why we have chosen to run version 2 of our algorithms with different values of τ , and to return the best result of these runs. Notice that even if the COLORING procedure is run 11 times in WP2, the total computing time remains very attractive since it is below 1 second for a network with $n = 1000$ mobiles (see Table 10).

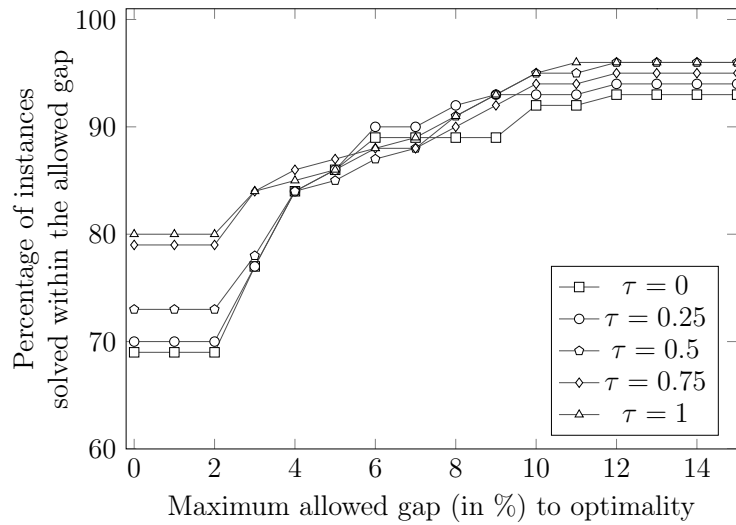


Figure 6: Performance profile of WP2 with different values of τ .

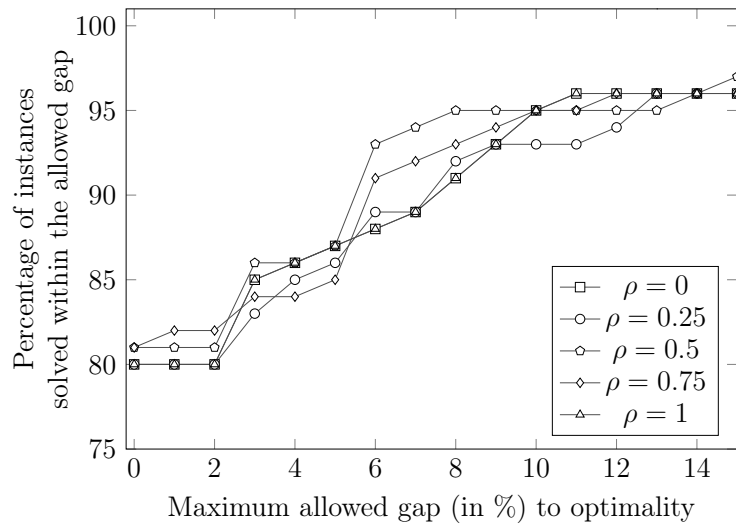


Figure 7: Performance profile of WP3 with different values of ρ .

Concerning parameter ρ , the performance profile in Figure 7 leads to the same conclusion: there seems to be no optimal setting for ρ , which explains why we run version 3 of our algorithms with eleven different values.

6.3 Comparisons with the FFR technique

We next compare the performance of the proposed heuristics with the Fractional Frequency Reuse (FFR) technique. It follows from Corollary 10 that the following value is an upper bound on the percentage of simultaneously communicating mobiles, when using the FFR technique:

$$\frac{100}{n} \sum_{p=1}^t (\min \{x_p, |F|\} + \min \{|F_p|, y_p, \max \{0, |F| - x_p\}\}).$$

We denote by FFR3 the value of this upper bound obtained by setting $|F_p| = \frac{|F|}{3}$, and $\tau = \frac{\theta}{2}$ for the definition of the center and the border of a cell (i.e., for determining x_p and y_p in each cell C_p). We find it

important to mention that FFR3 is a very optimistic upper bound on the performance of the FFR technique, not only because it assumes that mobiles in the center of a cell do not interfere with mobiles in adjacent cells, but also because real base station layouts do not permit to fix $|F_p| = \frac{|F|}{3}$ (see Figure 4).

Computational results for different pairs (t, k) are reported in Tables 11, 12, 13, 14, and 15. Each entry in these tables corresponds to the average coverage ratio (i.e., the percentage of mobiles to which a channel could be assigned) taken over 30 instances. Bounds on the optimal value OPT are given for instances having less than $t = 15$ base stations and $k = 48$ channels. More specifically, \underline{z} and \bar{z} are the best solution value and the best upper bound computed by CPLEX with a running time of 30 seconds. If $\underline{z} = \bar{z}$, it means that the CPLEX solver was able to reach optimality. Figure 8 shows the statistical distribution of these results for the triplets $(n, t, k) = (30, 10, 12)$, $(80, 10, 24)$, $(175, 15, 36)$ and $(225, 15, 48)$, with standard conventions for box-and-whisker plots.

Table 11: Coverage ratio with $(\theta, \gamma) = (0.25, 4)$, $t = 10$ base stations, $k = 12$ channels.

n	\underline{z}	\bar{z}	FFR3	WP1	WP2	WP3	DSAT1	DSAT2	DSAT3	RLF1	RLF2	RLF3
15	100.0	100.0	99.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
20	100.0	100.0	99.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
25	100.0	100.0	97.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
30	99.7	99.7	96.3	99.5	99.5	99.7	99.7	99.7	99.7	99.7	99.7	99.7
35	99.1	99.1	93.3	97.6	98.3	98.1	98.6	98.7	98.7	98.1	98.4	98.6
40	98.5	98.9	90.5	95.9	96.5	96.9	96.5	97.1	96.9	96.0	97.8	97.3
45	96.7	97.0	88.8	88.2	89.7	90.9	91.3	92.2	92.0	89.0	92.2	91.9
50	96.5	96.8	89.2	90.2	91.6	92.8	93.2	94.1	93.8	90.8	93.1	92.8

Table 12: Coverage ratio with $(\theta, \gamma) = (0.25, 4)$, $t = 10$ base stations, $k = 24$ channels.

n	\underline{z}	\bar{z}	FFR3	WP1	WP2	WP3	DSAT1	DSAT2	DSAT3	RLF1	RLF2	RLF3
60	99.7	99.8	95.1	99.3	99.4	99.5	99.4	99.6	99.7	99.2	99.4	99.3
65	99.8	99.8	95.6	99.5	99.5	99.5	99.7	99.7	99.7	99.4	99.5	99.5
70	100.0	100.0	96.2	99.8	100.0	100.0	99.9	100.0	100.0	99.9	100.0	100.0
75	99.5	99.7	96.3	98.1	98.6	99.0	98.9	99.3	99.3	98.5	98.9	98.9
80	99.6	99.8	94.9	97.8	98.1	98.4	98.8	98.9	99.1	98.3	99.0	98.9
85	98.9	99.5	93.6	97.0	97.3	97.8	97.3	97.8	97.6	97.4	97.9	98.1
90	98.4	99.0	91.3	94.0	94.6	95.4	95.3	96.1	96.4	94.3	96.2	96.3
95	97.3	98.5	90.1	92.7	94.1	94.9	94.9	96.0	96.1	93.2	95.2	95.2
100	97.2	98.6	89.7	91.3	92.9	94.7	94.3	95.7	96.0	91.8	94.5	95.1
105	92.9	96.3	89.4	84.2	85.9	88.5	88.8	90.2	89.8	85.2	88.6	88.7

Table 13: Coverage ratio with $(\theta, \gamma) = (0.25, 4)$, $t = 15$ base stations, $k = 36$ channels.

n	\underline{z}	\bar{z}	FFR3	WP1	WP2	WP3	DSAT1	DSAT2	DSAT3	RLF1	RLF2	RLF3
75	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
100	100.0	100.0	99.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
125	99.9	100.0	98.7	99.9	99.9	99.9	100.0	100.0	100.0	99.9	99.9	100.0
150	99.2	99.8	94.8	98.5	98.9	99.2	99.1	99.6	99.5	99.0	99.2	99.3
175	98.3	99.4	94.2	96.0	96.5	97.2	97.0	97.8	98.0	96.3	98.0	97.6
200	95.0	99.9	93.2	92.6	93.4	94.9	92.7	94.5	95.3	93.3	95.3	95.7
225	88.7	100.0	89.6	80.4	82.6	88.1	84.9	87.2	88.6	81.1	85.7	89.4

Table 14: Coverage ratio with $(\theta, \gamma) = (0.25, 4)$, $t = 15$ base stations, $k = 48$ channels.

n	\underline{z}	\bar{z}	FFR3	WP1	WP2	WP3	DSAT1	DSAT2	DSAT3	RLF1	RLF2	RLF3
75	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
100	100.0	100.0	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
125	100.0	100.0	99.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
150	100.0	100.0	98.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
175	100.0	100.0	98.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
200	99.7	100.0	97.9	99.6	99.7	99.8	99.8	99.9	99.9	99.8	99.8	99.8
225	95.0	100.0	95.2	95.3	96.0	97.1	96.7	97.0	97.1	96.2	97.5	97.2
250	0.0	100.0	93.2	92.1	92.8	94.1	93.7	94.3	94.8	92.7	94.7	94.4

Table 15: Coverage ratio with $(\theta, \gamma) = (0.25, 4)$, $t = 25$ base stations, $k = 120$ channels.

n	FFR3	WP1	WP2	WP3	DSAT1	DSAT2	DSAT3	RLF1	RLF2	RLF3
500	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
550	99.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
600	99.4	99.9	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0
650	99.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
700	98.9	99.5	99.7	99.8	99.8	99.9	99.9	99.4	99.7	99.9
750	99.0	99.7	99.8	99.9	99.7	99.9	100.0	99.8	99.9	100.0
800	97.9	98.1	98.7	99.5	98.2	98.9	99.9	98.2	99.0	99.7
850	98.1	98.4	98.6	99.5	98.2	98.6	99.2	98.2	98.7	99.6
900	97.0	95.8	96.5	98.9	95.1	96.4	98.3	95.9	96.9	99.2
950	96.9	93.8	94.8	98.5	92.5	94.7	98.9	93.1	95.8	98.8
1000	95.5	90.2	91.3	97.4	90.3	92.5	97.1	88.2	91.5	98.0
1050	94.2	83.5	84.7	93.9	83.6	85.9	90.8	82.0	87.6	94.6
1100	92.8	78.8	80.2	91.6	79.5	82.2	87.9	76.8	81.0	92.9

For networks having $t = 10$ base stations and $k = 12$ available channels, we observe in Table 11 that WP_i , $DSAT_i$ and RLF_i ($i = 1, 2, 3$) not only produce a better coverage ratio than the optimistic upper bound FFR3, but are also close to optimality. For example, with $n = 30$ mobiles, the optimistic upper bound FFR3 gives a coverage ratio of 96.3%, while the heuristics all produce a solution with at least 99.5% of communicating users, the optimal value being equal to 99.7%. For $n = 200$ mobiles and $(t, k) = (15, 48)$ (see Table 14), all the heuristics increase the coverage ratio FFR3 of 97.9% by approximately 2%, which corresponds to 4 additional mobiles that can communicate simultaneously. Similar conclusions hold for larger instances. For example, with $n = 800$ mobiles and $(t, k) = (25, 120)$ (see Table 15), FFR3 produces a coverage ratio of 97.9%, while the heuristics give coverage ratios ranging from 98.1% (WP1) to 99.9% (DSAT3), i.e. up to 16 additional mobiles that can communicate simultaneously. The statistical distributions in Figure 8 show that these results are stable.

A more visual comparison is given in Figure 9. For better readability, only the third versions WP3, DSAT3 and RLF3 of our algorithms are plotted. The best upper bound \bar{z} is omitted when it remains close to 100%. Since a coverage ratio of 98% corresponds to the typical accepted call-blocking probability (CBP) of 2%, we represent it with a horizontal line.

For $(t, k) = (10, 12)$, we observe that the CBP of 2% is reached by the FFR technique with 25 users, while more than 10 additional users can communicate with WP3, DSAT3 and RLF3. For $(t, k) = (15, 48)$, the number of communicating users with the CBP of 2% increases from 200 with the FFR technique to 220 with the three proposed heuristics.

The results obtained for realistic wireless networks with up to $t = 25$ base stations and $k = 120$ channels lead to interesting observations (see Table 15 and Figure 9). We first note that the coverage ratio of 98% is out of reach by the FFR technique with more than 850 mobiles, while at least 100 additional mobiles can communicate using heuristics WP3, DSAT3, or RLF3. If we replace $|F_p| = \frac{|F|}{3}$ by $|F_p| = \frac{|F|}{4}$, we get a more realistic upper bound for the FFR technique. The advantage of our heuristics is then even more impressive

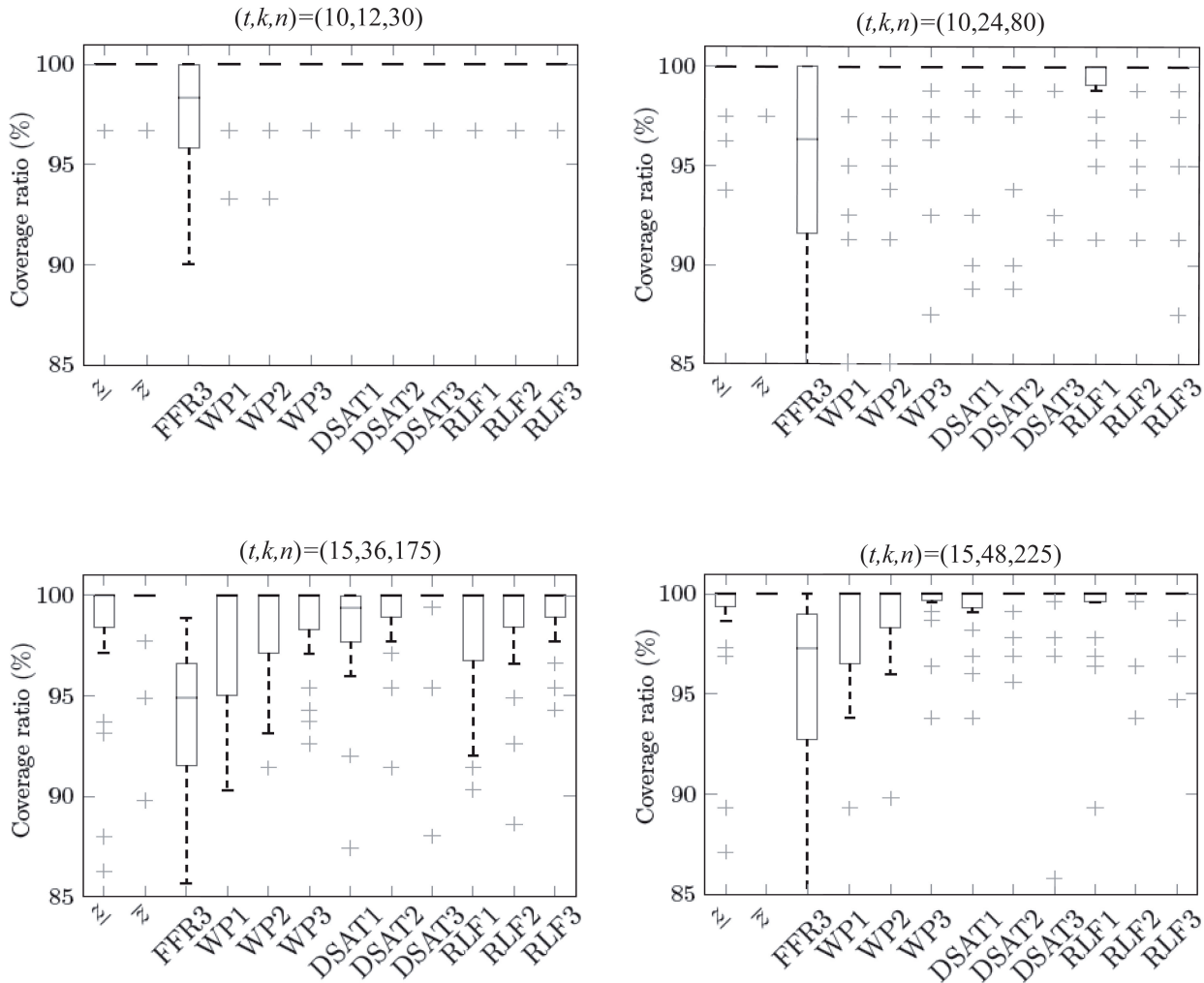


Figure 8: Boxplots of the coverage ratio with $(\theta, \gamma) = (0.25, 4)$.

since the coverage ratio of 98% is out of reach by the FFR technique with more than 650 mobiles, while 1000 mobiles can communicate using RLF3, which corresponds to an increase of more than 50%.

Observe that versions 1 and 2 of our algorithms are not as efficient as version 3 when the optimal coverage ratio draws away from 100%. For example, while the coverage ratio of WP3 for 1000 users is 91.6%, it lies between 78.8% and 80.2% for WP1 and WP2. Such a decrease of more than 10% represents more than 100 users. This can be explained as follows. When the optimal coverage ratio is strictly smaller than 100%, it seems reasonable to remove from the network those users having high interference with other mobiles. By keeping these users in the network, one takes the risk of assigning to them the same channel, thus preventing many other users from communicating. Heuristics WP3, DSAT3, and RLF3 were implemented so that users that interfere much with each other have a smaller chance to get the same channel.

7 Conclusion

Today's wireless networks should allow as many users as possible to communicate simultaneously, without compromising the quality of the signals. We have shown that such a goal can be achieved in an efficient way by solving the Partial Directed Weighted Improper Coloring Problem (PDWICP). Wireless networks must ensure a call blocking probability (CBP) below 2% (i.e., a coverage ratio of 98%) to satisfy the current mobile standards. We have adapted three standard graph coloring algorithms to the PDWICP, with three versions for

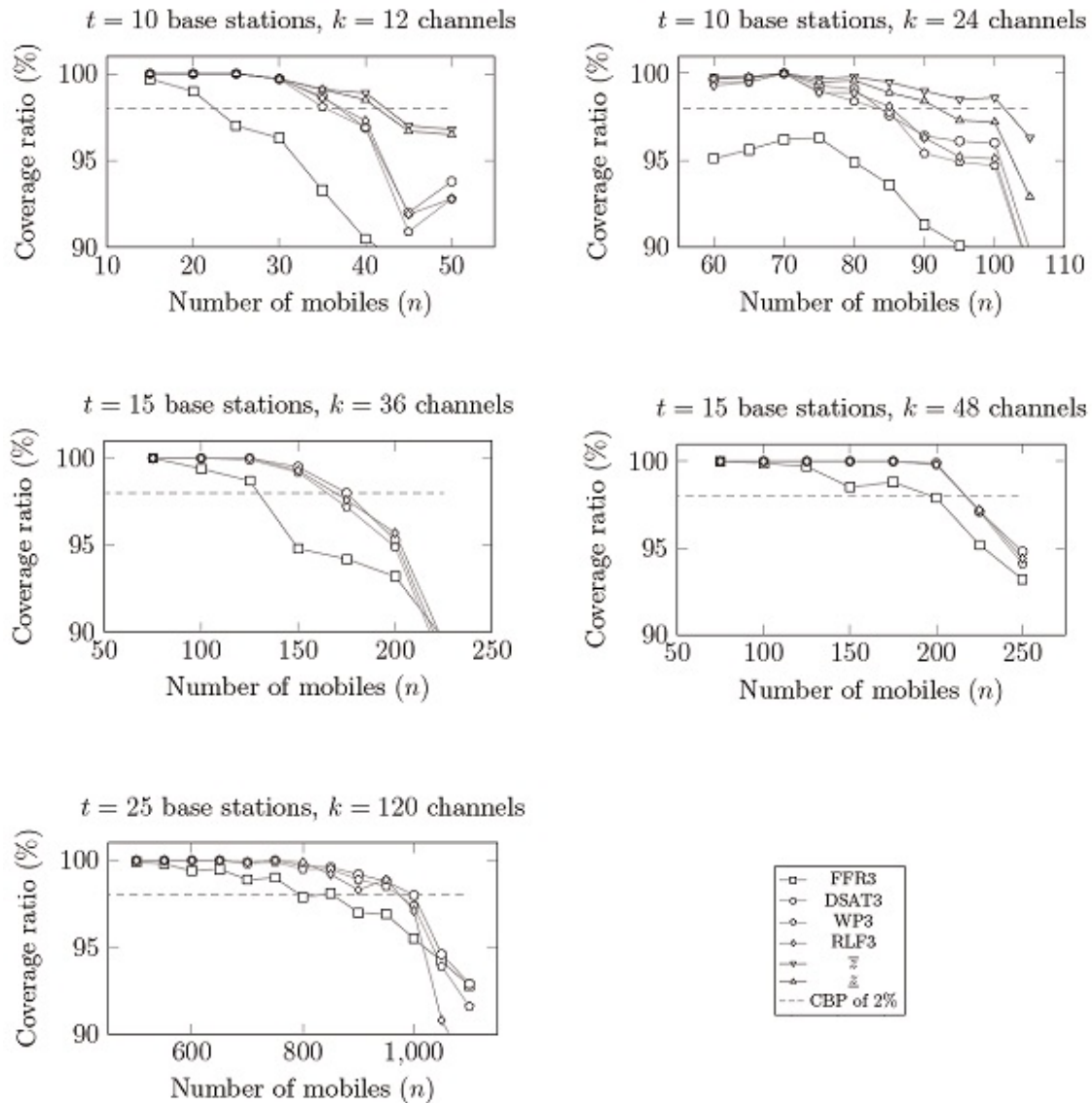


Figure 9: Evolution of the coverage ratio with $(\theta, \gamma) = (0.25, 4)$.

each algorithm. All proposed heuristics meet the CBP requirement of 2% with more users than the standard FFR technique. In summary, the proposed heuristics perform better than today’s phone operator systems that mostly use decentralized channel assignment strategies. In particular the low complexity of the WP3 algorithm, when compared to DST3 or RLF3, makes it very attractive for real-time channel assignments.

We have also noticed that the performance of the proposed algorithms decreases when the optimal coverage ratio draws away from 100%. This can be explained by the fact that our heuristics make the implicit assumption that all vertices will be colored. Indeed, in their original form, the WP, DSAT and RLF algorithms aim at finding the chromatic number of a graph. For example the WP algorithm sequentially colors the vertices by non increasing degree, in order to give priority to vertices having many neighbors. If it is known in advance that the size of the largest induced k -colorable subgraph is much smaller than n , it then seems reasonable to sort the vertices in non decreasing degree to exclude in priority vertices that are likely to saturate color classes. Solving the PDWICP in situations where the optimal coverage ratio is much below 100% would therefore probably require readapting the order in which vertices are considered for a sequential coloring.

References

- [1] C. Archetti, N. Bianchessi, A. Hertz, A. Colombet, and F. Gagnon, Directed weighted improper coloring for cellular channel allocation, *Discrete Applied Mathematics*, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.dam.2013.11.018>
- [2] J. Araujo, J.-C. Bermond, F. Giroire, F. Havet, D. Mazauric, and R. Modrzejewski, Weighted improper colouring, in *Proceedings of the 22nd international conference on Combinatorial Algorithms*, ser. IWOCA'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 1–18. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25011-8_1
- [3] F. Aurenhammer, Voronoi diagrams – a survey of a fundamental geometric data structure, *ACM Computing Surveys*, 23(3), 345–405, 1991.
- [4] G. Boudeau, J. Panicker, N. Guo, R. Chang, N. Wang, and S. Vrzic, Interference coordination and cancellation for 4g networks, *IEEE Communications Magazine*, 47(4), 74–81, 2009.
- [5] K. Appel and W. Haken, Solution of the four color map problem, *Scientific American*, 237(4), 108–121, 1977.
- [6] D.J.A. Welsh and M.B. Powell, An upper bound for the chromatic number of a graph and its application to timetabling problems, *The Computer Journal*, 10(1), 85–86, 1967.
- [7] D. Brélaz, New methods to color the vertices of a graph, *Communications of the ACM*, 22(4), 251–256, 1979.
- [8] F.T. Leighton, A graph coloring algorithm for large scheduling problems, *Journal of research of the national bureau of standards*, 84(6), 489–506, 1979.
- [9] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, The four-colour theorem, *Journal of Combinatorial Theory, Series B*, 70, 2–44, 1967.