CONSTRAINT AND INTEGER PROGRAMMING MODELS FOR BANDWIDTH COLORING AND MULTICOLORING IN GRAPHS

Bruno Dias, Rosiane de Freitas
Instituto de Computação - Universidade Federal do Amazonas
Manaus/AM - Brazil
{bruno.dias, rosiane}@icomp.ufam.edu.br

Nelson Maculan
PESC/COPPE - Universidade Federal do Rio de Janeiro
Rio de Janeiro/RJ - Brazil
maculan@cos.ufrj.br

Philippe Michelon
Centre d'Enseignement et de Recherche en Informatique, Université d'Avignon et des Pays de Vaucluse
Avignon - France
philippe.michelon@univ-avignon.fr

ABSTRACT

In this paper, constraint and integer programming formulations are applied to solve Bandwidth Coloring Problem (BCP) and Bandwidth Multicoloring Problem (BMCP). The problem definitions are explored in order to construct the constraint programming formulation. The integer programming formulation is based on a previous formulation for the related Minimum Span Frequency Assignment Problem (MS-FAP), which is modified in order to reduce its size and computation time. The two exact approaches are implemented with available solvers and applied to well-known sets of instances from the literature, GEOM and Philadelphia-like problems. Using these models, some heuristic solutions from previous works are proven to be optimal, a new upper bound for an instance is given and all optimal solutions for the Philadelphia-like problems are presented. A discussion is also made on the performance of constraint and integer programming for each considered coloring problem, and the best approach is suggested for each one of them.

KEYWORDS. Graph coloring, integer programming, constraint programming.

1. Introduction

A *k-coloring* of a graph $G = (V, E)$, with V as its set of vertices and E as its set of edges, is an assignment $x : V \to \{1, 2, \ldots, k\}$ such that $\forall (i, j) \in E,\ x(i) \neq x(j)$. The *chromatic number* χ_G of a graph is the minimum value of k for which G is *k-colorable*. The classic graph coloring problem, which consists in finding the chromatic number of a graph, is a classical combinatorial optimization problem which belongs to NP-hard complexity class (Karp, 1972).

One of the main applications of such problems consists of assigning channels to transmitters in a mobile wireless network. Such assignment of the channels cannot be made in an arbitrary way, since there is the problem of interference among devices located near each other using approximate channels. Given this scenario, channels must be assigned to calls so interference is avoided and the spectrum usage is minimized (Koster, 1999; Audhya et al., 2011; Gokbayrak and Yıldırım, 2013).

A useful generalization of the classic graph coloring problem for channel assignment is Bandwidth Coloring Problem (BCP) Phan and Skiena (2002); Malaguti and Toth (2008, 2010); Lai and Lü (2013), where, for each edge $(i, j) \in E$, there is a positive integer $d_{i,j}$ and a coloring is a mapping $x : V \to \mathbb{N}$ such that $\forall (i, j) \in E,\ |x(i) - x(j)| \geq d_{i,j}$. These $d_{i,j}$ values represent

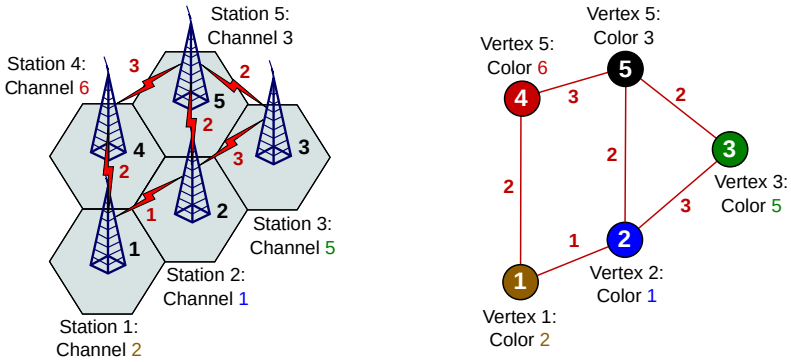


Figure 1: Example of channel assignment and its modeling as an instance of the bandwidth coloring problem.

the minimum separation needed between channels of nearby stations in order to avoid harmful interference. An example of channel assignment and its corresponding bandwidth coloring model is shown in Figure 1.

The main contribution of this paper consists of the use of integer and constraint programming models to provide exact solutions to BCP, applying them to existing instances, including ones based on real channel assignment scenarios. There are many algorithms for BCP in the literature, including some based on classic metaheuristics, including simulated annealing Costa (1999); Dias et al. (2013), local search Galinier and Hertz (2006), evolutionary algorithms Malaguti and Toth (2008) and tabu search Dorne and Hao (1998); Lai and Lü (2013). However, there is no optimality guarantee in these methods. Using integer and constraint programming approaches, we were able to prove the optimality of some solutions found by heuristic methods, such as the multistart iterated tabu search proposed in Lai and Lü (2013), and obtain better upper bounds for some problems, including optimal solutions for open instances. In this process, we also found some inconsistencies in the literature with respect to the quality of some approximate algorithms, where the heuristic presented solutions better than the optimal ones found by an exact method.

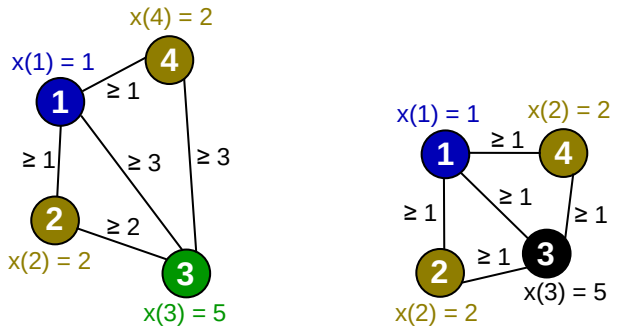
The remainder of this paper is organized as follows. Section 2 formally defines the Bandwidth Coloring Problem and discusses its characteristics. Section 3 gives the constraint and integer programming models used to solve BCP and some variations. Section 4 shows results of some experiments done with implementations of the mathematical models. Finally, in Section 5, final remarks are made and next steps of ongoing research are stated.

2. Bandwith Coloring definitions and models

Bandwidth Coloring Problem (BCP) can be stated as follows. Given an undirected graph $G = (V, E)$ where, for each edge $(i, j) \in E$, there is a positive integer $d_{i,j}$, each vertex i must receive a color $x(i)$ and, for each edge $(i, j) \in E$, the condition $|x(i) - x(j)| \geq d_{i,j}$ must hold. An instance of the classic graph coloring problem can be transformed to BCP by considering $\forall (i, j) \in E,\ d_{i,j} = 1$. Examples of BCP are given in Figure 2.

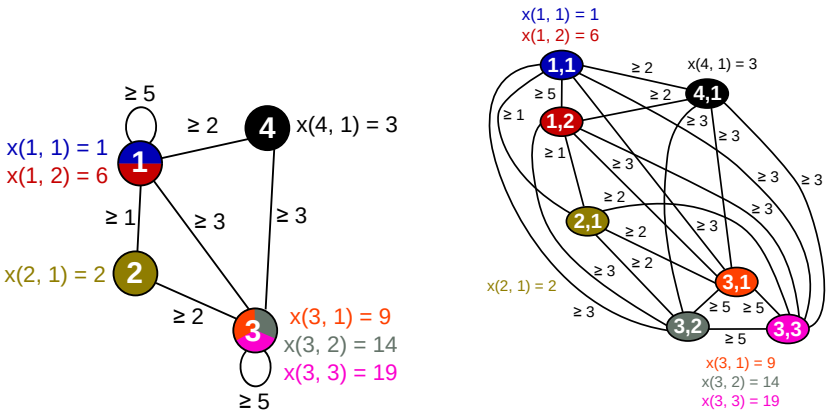
This problem is a special case of the T-coloring problem Hale (1980), where, for each edge, there is a set $T_{i,j}$ such that $|x(i) - x(j)| \notin T_{i,j}$. We can build a T-coloring instance from any BCP instance by setting the forbidden set of an edge $(i, j) \in E$ to $T_{i,j} = \{0, 1, \ldots, d_{i,j}\}$. The constraint $|x(i)-x(j)| \notin T_{i,j}$ is, then, the same as the one from BCP, that is, the former corresponds to a T-coloring instance with forbidden sets consisting of consecutive values.

A variation of BCP is the Bandwidth Multicoloring Problem (BMCP), where the vertex i has an associated color demand q_i and a weight $d_{i,i}$, such that it receives q_i colors (instead of just one). Let $x(i, k)$ be the k-th color assigned to vertex i (with $1 \leq k \leq q_i$). Then, for each pair of colors $x(i, k)$ and $x(i, m)$ associated to i, the constraint $|x(i, k) - x(i, m)| \geq d_{i,i}$ must be respected in BMCP. An equivalent problem to BMCP is the minimum span frequency assignment (MS-FAP) Koster (1999); Audhya et al. (2011), where channels correspond to colors and devices



(a) BCP. (b) Classic coloring.

Figure 2: Examples of instances for BCP and classic graph coloring (viewed as BCP).



(a) Original BMCP instance. (b) Transformation to BCP.

Figure 3: Example of BMCP instance and its transformation into a BCP instance (with unitary color demands).

to vertices. However, the input consists of positions for these devices and reuse distances, where, based on the distance between two devices, the separation between colors is calculated. If this input is converted into a graph where edges are weighted according to this separation, then it becomes a BMCP instance.

As is the case with all multicoloring problems, there is an equivalence between BCP and BMCP, that is, an instance of BMCP can be converted into BCP by replicating each vertex i into a clique of q_i subvertices. Each edge in the clique has a weight equal to $d_{i,i}$ from the original BMCP instance, and each subvertex is adjacent to all other vertices that the original vertex also was adjacent to. Figure 3 shows an example of this conversion.

3. Constraint and integer programming approaches

In order to obtain exact solutions for BCP and BMCP, we can model these problems using discrete optimization approaches such as constraint and integer programming, which can be then applied to a solver.

The first formulation we explore is based on constraint programming (CP). Let x_i be an integer variable consisting of the color assigned to vertex i. The CP model is then:

$$\text{Minimize} \quad \max_{i \in V} x(i) \tag{1}$$

$$\text{Subject to} \quad |x(i) - x(j)| \geq d_{i,j} \quad (\forall (i,j) \in E) \tag{2}$$

$$x(i) \in \mathbb{Z}^* \quad (\forall i \in V) \tag{3}$$

The objective (1) is to minimize the maximum used color among all vertices (the coloring span). Constraint set (2) involves weighted edges with inequality constraints. For each variable x_i, the

initial domain (before constraint propagation) $D(x_i)$ consists of all integers between 1 and a given upper bound U, that is, $D(x_i) = [\![1, U]\!]$. Note that all initial domains are the same. This model has $O(|V|)$ variables (one for each vertex) and $O(|E|)$ constraints (one for each edge), since it captures the essential definition of the problem.

A special CP model can be defined for the case when all edge weights are the same (for example, for classic graph coloring). Let β be a natural number equivalent to the weight imposed on all edges. Using a specific global constraint, we state the formulation as shown below:

$$\text{Minimize} \quad \max_{i \in V} x(i) \tag{4}$$

$$\text{Subject to} \quad \text{allMinDistance}(\{x(j) : (i,j) \in E\},\ \beta) \quad (\forall i \in V) \tag{5}$$

$$x(i) \in \mathbb{Z}^* \quad (\forall i \in V) \tag{6}$$

The *allMinDistance* global constraint used in 5 takes as its arguments a set of variables and a constant w, and ensures that, for all pairs of variables y and z in the set, the condition $|y - z| \geq w$ is valid, which is the case for each vertex and its neighbors. This special case has $O(|V|)$ variables and also $O(|V|)$ constraints. Since, for most connected graphs, we have $|V| \leq |E|$, this formulation has fewer constraints and is able to use specialized propagators in the search.

For BMCP, a CP formulation can be constructed by using characteristics from both previously shown models. As discussed in Section 2, a multicoloring problem can be converted into a coloring with single demands by transforming a vertex i into a clique of q_i vertices, each adjacent to all other vertices that were adjacent to i. By using this, we have that, essentially, each vertex consists of a small BCP subinstance with all edge weights being the same, where the larger graph (that is, considering the constraints imposed on the original edges between different vertices), if its multicoloring demands are ignored, is a BCP instance. This combination leads to the following CP formulation:

$$\text{Minimize} \quad \max_{\substack{i \in V \\ 1 \leq k \leq q_i}} x(i,k) \tag{7}$$

$$\text{Subject to} \quad |x(i,k) - x(j,m)| \geq d_{i,j} \quad (\forall (i,j) \in E,\ 1 \leq k \leq q_i,\ 1 \leq m \leq q_j) \tag{8}$$

$$\text{allMinDistance}(\{x(i,k) : 1 \leq k \leq q_i\},\ d_{i,i}) \quad (\forall i \in V) \tag{9}$$

$$x(i,k) \in \mathbb{Z}^* \quad (\forall i \in V,\ 1 \leq k \leq q_i) \tag{10}$$

In this model, constraints (8) ensure that colors assigned to different vertices respect the weight imposed on edges. Constraints (9) require that different colors assigned to the same vertex respect the minimum absolute difference $d_{i,i}$ between each other (using the *allMinDistance* global constraint, since they form a small BCP subinstance with all edge weights being the same). This formulation has $O(|V| \times q_{max})$ variables (where $q_{max} = \max_{i \in V'} q_i$, that is, the largest color demand in the graph), since, for each color needed to each vertex, there is a variable, and $O(|E| \times q_{max})$ constraints.

The second mathematical formulation approach is an integer programming model, as proposed independently in Dias (2014) and Mak (2007) based on models defined in Koster (1999). Two sets of variables are used:

- $x_{ic} = \begin{cases} 1 & \text{if color } c \text{ is assigned to vertex } i; \\ 0 & \text{otherwise.} \end{cases}$
- z_{max} = value of maximum color used in the solution (the coloring span).

Using these variables, the IP model is defined as follows:

$$\text{Minimize} \quad z_{max} \tag{11}$$

$$\text{Subject to} \quad \sum_{c=1}^{U} x_{ic} = 1 \quad (\forall i \in V) \tag{12}$$

$$x_{ic} + x_{je} \leq 1 \quad (\forall (i,j) \in E;\ 1 \leq c, e \leq U \mid |c-e| < d_{i,j}) \tag{13}$$

$$z_{max} \geq c x_{ic} \quad (\forall i \in V;\ 1 \leq c \leq U) \tag{14}$$

$$x_{ic} \in \{0,1\} \quad (\forall i \in V;\ 1 \leq c \leq U) \tag{15}$$

$$z_{max} \in \mathbb{R} \tag{16}$$

The objective (11) is to minimize the value of variable z_{max}, which will be the coloring span. Constraint set (12) ensures that all vertices must be colored. Constraint set (13) ensures that the absolute difference between the colors assigned to adjacent vertices is less than the weight of the respective edge, then only one of the vertices can receive that color. Constraints (14) require that variable z_{max} be greater than or equal to any color used, so it will be the maximum color used. Constraints (15) require that variables in the set x_{ic} use only values 0 and 1, while constraint (16) states that z_{max} is a free variable, although its value will always be an integer, since, at the optimal solution, $z_{max} = c x_{ic}$ for some $i \in V$ and $c \in [\![1, U]\!]$. The value U denotes the upper bound for colors to be used, since variables are indexed by vertex and color, so the color set must be limited. This IP model has $O(U \times |V|)$ variables (one for each pair of color and vertex) and $O(U \times (|E| + |V|))$ constraints, that is, it has pseudopolynomial size.

For BMCP, the integer programming model can also be modified to accomodate multi-coloring demands. Two modifications must be made for the new model, The first one is changing the RHS of constraints (12) from 1 to q_i, which ensures that, instead of receiving only one color, each vertex i will receive q_i colors. The second one is expanding the set of constraints (13) in order to add new ones for ensuring that the same vertex i avoids using colors that violate the weight $d_{i,i}$, that is, there will be one constraint for each $(i,j) \in E;\ 1 \leq c, e \leq U$ such that $|c-e| < d_{i,j}$ and also for each $i \in V;\ 1 \leq c, e \leq U$ such that $|c-e| < d_{i,i}$. Note that this is equivalent to having an edge (i,i) for each vertex i, which would make the new constraints be automatically included in the original constraint set. These modifications do not impact the asymptotic size of the formulation. The full model is given below:

$$\text{Minimize} \quad z_{max} \tag{17}$$

$$\text{Subject to} \quad \sum_{c=1}^{U} x_{ci} = q_i \quad (\forall i \in V) \tag{18}$$

$$x_{ic} + x_{je} \leq 1 \quad (\forall (i,j) \in E;\ 1 \leq c, e \leq U \mid |c-e| < d_{i,j}) \tag{19}$$

$$x_{ic} + x_{ie} \leq 1 \quad (\forall i \in V;\ 1 \leq c, e \leq U \mid |c-e| < d_{i,i}) \tag{20}$$

$$z_{max} \geq c x_{ic} \quad (\forall i \in V;\ 1 \leq c \leq U) \tag{21}$$

$$x_{ci} \in \{0,1\} \quad (\forall i \in V;\ 1 \leq c \leq U) \tag{22}$$

$$z_{max} \in \mathbb{R} \tag{23}$$

3.1. Upper bounds for color sets

Both CP and IP models need a finite color set, which, as shown previously, consist of an interval of integers $[\![1, U]\!]$, where U is an upper bound for the coloring span. A trivial value for U can be calculated by summing the weights imposed on all edges, that is, $U = \sum_{(i,j) \in E} d_{i,j}$. However, this upper bound is very weak, since, by summing all weights, we are ignoring the fact that colors can be reused by vertices not adjacent to each other, which makes the coloring span become a large

Algorithm 1 Greedy heuristic for generating starting solutions for BCP and BMCP.

Require: graph G (with set V of vertices and set E of edges), weights $d : E \cup \{(i,i) : i \in V\} \rightarrow \mathbb{Z}_{\geq 0}$, color demands $w : V \rightarrow \mathbb{N}$.

function GREEDYSTARTINGSOLUTION($G = (V, E), d, w$)
  $V' \leftarrow V$
  for each $i \in V$ **do**
    $numCol[i] \leftarrow 0$
    $colorAssign[i] \leftarrow \emptyset$
  while $V' \neq \emptyset$ **do**
    $i \leftarrow \arg\max_{v \in V'} q_v$
    $candColor \leftarrow (numCol[i] \times d_{i,i}) + 1$
    for each $j \in V' - \{i\}$ **do**
      for each $k \in colorAssign[v]$ **do**
        if $|k - candColor| < d_{i,j}$ **then**
          $candColor \leftarrow candColor + 1$
    $colorAssign[i] \leftarrow colorAssign[i] \cup \{candColor\}$
    $V' \leftarrow V' - \{i\}$
  return $colorAssign$

value far from optimum. This also makes the color set have a large cardinality, which has a huge impact on the computing performance of these models, especially the IP model, since the number of variables and constraints are proportional to the upper bound, which can lead to out-of-memory scenarios.

A better value for U is given by preprocessing the input graph, where an heuristic which does not need an explicit upper bound is applied to it. The span of the resulting solution is used as the value U when assembling CP and IP models for the input graph. A greedy algorithm for coloring the input graph can be used for this, where the vertices are processed following an order based on their color demands - vertices with higher demands are colored first. A color for a vertex i is determined by first setting its color candidate as $numCol[i] \times d_{i,i} + 1$, where $numCol[i]$ is the number of colors already assigned to i, and checking if it violates any separation constraint with any of its neighbor vertices. If a violation occurs, the color candidate is incremented by 1 and this checking is made again until a feasible color is found. The color is then assigned to i and, if its demands are not fully satisfied, an additional color is calculated and assigned to it. This is repeated until the graph is fully colored. Algorithm 1 gives pseudocode for this greedy heuristic. The upper bound is, then, the span of the solution returned by this method.

4. Computational experiments

The constraint and integer programming formulations were implemented in C++ using IBM/ILOG CPLEX solver, version 12.5.1, and its CP Optimizer component. The resulting programs were executed in a Microsoft Azure A9 Virtual Machine, consisting of Intel Xeon E5-2670 processors (16 cores @ 2.6 GHz), 112GB of RAM and Ubuntu Linux 14.04.1 LTS operating system. Both formulations used the standard parameters of the solver, but using only one thread, and each instance was limited to 48 hours of CPU time (172800 seconds).

We used two sets of instances from the literature in our experiments. The first set of literature instances is known as GEOM, generated by Michael Trick Trick et al. (2002) for BCP and its multicoloring variant, BMCP, and consists of 33 instances of three types: GEOMn are sparse graphs, while GEOMna and GEOMnb are denser graphs (where n is the number of vertices in the instance).

The second set of instances consists of the classical phia (21 stations) and Helsinki (25 vertices) problems for MS-FAP, based on cellular networks from the cities of the same names,

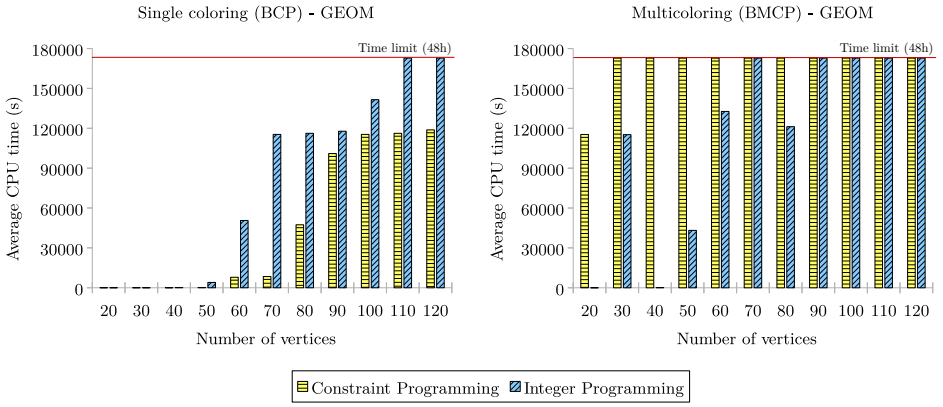


Figure 4: Number of vertices × CPU time needed to prove optimality (if found) for each method on GEOM instances.

and an artificial problem (55 vertices) that extends a Philadelphia instance, as seen in Chakraborty (2001); Dias et al. (2013); Kendall and Mohamad (2005). Since these instances are defined for BMCP only, we applied the suggestion by the authors of the GEOM instances in these as well to generate BCP instances.

When constructing the models for each instance, we executed the preprocessing discussed in Subsection 3.1 in order to obtain a feasible solution and an upper bound. To further help the solvers, we fed the entire starting solution to them, namely, we passed the solution as a starting point to CP Optimizer and as a MIP start to CPLEX, instead of only using the span as an upper bound. This is especially important for CPLEX, since it guarantees that an optimality gap is calculated as soon as the enumeration starts.

The first results presented are for BCP. Table 1 shows the results for the GEOM BCP instances. Since the BCP variants are also used in the literature, we compared our results with the Discropt heuristic framework in Phan and Skiena (2002) and the multistart iterated tabu search heuristic in Lai and Lü (2013) to verify the correctness of the solutions by the CP and IP formulations. For all sparse instances (the ones without 'a' or 'b' in the name), the constraint programming implementation was able to prove optimality. However, we emphasize that, for some instances, no method achieved the best solution presented in Phan and Skiena (2002). As noted in Lai and Lü (2013), no other work has obtained the same results, while our exact approaches reached the same best solutions for these instances obtained by other authors, which leads us to believe there is a mistake in Phan and Skiena (2002), as marked in Table 1.

Table 2 shows the results for MS-FAP (Philadelphia, Helsinki and Artificial) instances without considering multicoloring demands. We note that, for each Philadelphia constraint matrix C^i_{21} where i is odd, by dropping the multicoloring demands, the instances become equal, since the only difference among them is the separation between colors of the same vertex. The same occurs for each even i. Given that, we grouped together results for each odd i (1, 3, 5 and 7) and for each even i (2, 4, 6, 8). Again, the CP formulation reaches optimality for each instance much faster, although runtimes are small due to the size of these relaxed instances.

The next experiments are for BMCP. For these instances, a lower bound can be calculated as $L = \max[(d_{i,i} \times (q_i - 1)) + 1]$, as shown by Chakraborty (2001). This value is also inserted in the models by adding a single constraint requiring that the objective value be greater than or equal to L, which helps the enumeration end faster when the optimal solution has span equivalent to this lower bound, especially when using CP.

Table 3 shows the results obtained for the GEOM instances. We compared our results obtained with such instances with the same multistart iterated tabu search heuristic from Lai and Lü (2013), where the algorithm for BCP is applied to BMCP instances by expanding the vertices

Table 1: Results for the constraint and integer programming formulations applied on literature BCP instances (GEOM set). Underlined results in "Best Found" columns are optimal values.

Instance	Num. Vert.	Phan and Skiena (2002)	Lai and Lu (2013)	Constr. Progr. (CP Optimizer)		Integer Progr. (CPLEX)	
		Best Reported	Best Reported	Best Found	Time (sec)	Best Found	Time (sec)
GEOM20		20*	21	21	0.00	21	0.33
GEOM20a	20	20	20	20	0.02	20	0.95
GEOM20b		13	13	13	0.01	13	0.09
GEOM30		27*	28	28	0.05	28	0.88
GEOM30a	30	27	27	27	0.05	27	8.06
GEOM30b		26	26	26	0.03	26	2.27
GEOM40		27*	28	28	0.05	28	1.97
GEOM40a	40	38	37	37	1.39	37	278.66
GEOM40b		36	33	33	2.06	33	252.39
GEOM50		29	28	28	0.26	28	21.44
GEOM50a	50	54	50	50	374.42	50	3457.25
GEOM50b		40	35	35	144.69	35	8494.52
GEOM60		34	33	33	1.12	33	45.73
GEOM60a	60	54	50	50	684.59	50	16755.65
GEOM60b		47	41	41	22915.94	41	134996.77
GEOM70		40	38	38	2.39	38	533.53
GEOM70a	70	64	61	61	24798.03	$\leq$ 62	172815.55
GEOM70b		54	47	47	534.65	$\leq$ 49	172834.40
GEOM80		44	41	41	8.18	41	3019.18
GEOM80a	80	69	63	63	87770.77	$\leq$ 65	172803.55
GEOM80b		70	60	60	54320.89	$\leq$ 66	172800.25
GEOM90		48	46	46	55.18	46	7768.62
GEOM90a	90	74	63	63	130050.12	None	173100.57
GEOM90b		83	69	$\leq$ 69	172800.00	None	172802.83
GEOM100		55	50	50	545.79	50	78836.94
GEOM100a	100	84	67	$\leq$ 70	172800.01	$\leq$ 85	172824.54
GEOM100b		87	72	**$\leq$ 71**	172800.02	$\leq$ 101	172840.38
GEOM110		59	50	50	2982.24	$\leq$ 50	172800.62
GEOM110a	110	88	72	$\leq$ 73	172800.01	$\leq$ 97	172811.66
GEOM110b		87	78	$\leq$ 79	172800.01	$\leq$ 99	172821.35
GEOM120		67	59	59	10778.18	None	173296.11
GEOM120a	120	101	82	$\leq$ 84	172800.01	None	173181.91
GEOM120b		103	84	$\leq$ 85	172800.01	None	173187.16

*Results lower than the obtained optimum - possibly wrong in the corresponding work.

Table 2: Results for the constraint and integer programming formulations applied on literature MS-FAP instances (Philadelphia, Helsinki and Artificial) without multicoloring demands.

Const. matrix	Num. Vert.	Constr. Progr. (CP Optimizer)		Integer Progr. (CPLEX - B&C)	
		Best Found	Time (sec)	Best Found	Time (sec)
C_{21}^1, C_{21}^3, C_{21}^5 and C_{21}^7	21	7	0.40	7	0.87
C_{21}^2, C_{21}^4, C_{21}^6 and C_{21}^8		9	0.06	9	2.66
C_{25}^1	25	8	4.71	8	1.90
C_{55}^1	55	7	0.79	7	30.63

into cliques as shown in Section 2, and also with the general purpose heuristic system (Discropt) used in Phan and Skiena (2002). We also used the MS-FAP (Philadelphia, Helsinki and Artificial) instances in their original forms (that is, including multicoloring) and compared with the constructive heuristic from Chakraborty (2001), the local search algorithm from Kendall and Mohamad (2005) and the memetic algorithm from Kim et al. (2007).

For BMCP, the efficiency between CP and IP is practically reversed: most instances are solved to optimality faster with IP than CP. In fact, for the MS-FAP instances, we were able to obtain all optimal values using IP. This is explained by taking into account, as discussed in Section 3, that the IP model is able to consider multicoloring demands without expanding the number of vertices, only having to change a set of constraints and add another set for the different colors for each vertex. However, the CP model has to grow more, since, essentially, a BMCP instance is treated as a special BCP one, since the number of variables increases and a new set of constraints is introduced. Figure 4 shows this difference in efficiency between methods. However, CP still has an advantage in BMCP: when it is unable to solve a problem to optimality in the given time limit, the solution that it returns has a better quality than the one found by IP (that is, the coloring span of the solution found by CP is lower than the one found by IP). This happens because CP has explicit information about which colors each vertex can assume, instead of calculating such colors.

We also detected a mistake in Chakraborty (2001), where the heuristic result presented in it for constraint matrix C_{25}^1 and demand vector D_{25}^4 is better (with objective function value 121) than the exact solutions obtained by both CP and IP (with objective function value 200). In fact, no other work in the literature obtained a solution with span lower than 200.

5. Concluding remarks

In this paper, we addressed the bandwidth coloring problem and its multicoloring variation, both of which have important applications for channel assignment in wireless networks. We employed constraint and integer programming formulations, which were implemented using computational OR tools, and applied them to instances from the literature and new random ones in order to verify which mathematical modeling tool is best for each coloring problem. Since the constraints from the problems are naturally transported to constraint programming, its implementation reaches the optimal solution much faster than the integer programming one.

Ongoing and future works include improving the CP formulation by domain reduction and more specific constraints, and also use hybrid methods, combining both CP and IP, as well as heuristics, in order to solve these models faster. The study of the problems posed to specific classes of graphs, in order to establish the characterization of feasibility conditions for them, is subject of the research in progress.

References

Audhya, G., Sinha, K., Ghosh, S., and Sinha, B. (2011). A survey on the channel assignment problem in wireless networks. *Wireless Communications and Mobile Computing*, 11:583–609.

Table 3: Results for the constraint and integer programming formulations applied on literature BMCP instances (GEOM set).

Instance	Num. Vert.	Lower Bound	Lai and Lu (2013)	Constr. Progr. (CP Optimizer)		Integer Progr. (CPLEX - B&C)	
			Best Reported	Best Found	Time (sec)	Best Found	Time (sec)
GEOM20	20	91	149	$\leq$ 149	172800.01	149	15.17
GEOM20a		91	169	$\leq$ 169	172800.01	169	18.49
GEOM20b		21	44	44	476.92	44	1.58
GEOM30	30	91	160	$\leq$ 160	172800.01	$\leq$ 160	172830.72
GEOM30a		91	209	$\leq$ 215	172800.01	$\leq$ 211	172813.47
GEOM30b		21	77	$\leq$ 77	172800.00	77	41.87
GEOM40	40	91	167	$\leq$ 168	172800.01	167	1192.28
GEOM40a		91	213	$\leq$ 225	172800.01	213	111262.08
GEOM40b		21	74	$\leq$ 74	172800.00	74	17027.77
GEOM50	50	91	224	$\leq$ 226	172800.02	224	52450.85
GEOM50a		91	314	$\leq$ 332	172800.03	$\leq$ 361	172820.13
GEOM50b		21	83	$\leq$ 85	172800.00	$\leq$ 87	172819.47
GEOM60	60	91	258	$\leq$ 259	172800.02	258	156987.80
GEOM60a		91	356	$\leq$ 380	172800.03	$\leq$ 448	172813.01
GEOM60b		21	113	$\leq$ 117	172800.01	$\leq$ 125	172897.07
GEOM70	70	91	270	$\leq$ 284	172800.03	$\leq$ 305	172804.56
GEOM70a		91	467	$\leq$ 483	172800.05	$\leq$ 578	172839.51
GEOM70b		21	116	$\leq$ 123	172800.01	$\leq$ 134	172805.88
GEOM80	80	91	381	$\leq$ 395	172800.04	$\leq$ 511	172809.70
GEOM80a		91	361	$\leq$ 382	172800.05	$\leq$ 479	172885.02
GEOM80b		21	139	$\leq$ 145	172800.01	$\leq$ 170	172820.56
GEOM90	90	91	330	$\leq$ 342	172800.05	$\leq$ 423	172811.82
GEOM90a		91	375	$\leq$ 392	172800.07	$\leq$ 452	172830.60
GEOM90b		21	144	$\leq$ 156	172800.01	$\leq$ 212	172844.07
GEOM100	100	91	404	$\leq$ 426	172800.07	$\leq$ 493	173190.69
GEOM100a		91	442	$\leq$ 465	172800.08	$\leq$ 596	172871.84
GEOM100b		21	156	$\leq$ 169	172800.02	$\leq$ 220	172810.33
GEOM110	110	91	381	$\leq$ 399	172800.07	$\leq$ 500	172840.98
GEOM110a		91	488	$\leq$ 527	172800.10	$\leq$ 610	173095.42
GEOM110b		21	204	$\leq$ 207	172800.01	$\leq$ 250	172835.82
GEOM120	120	91	396	$\leq$ 427	172800.06	$\leq$ 505	172923.18
GEOM120a		91	554	$\leq$ 585	172800.16	$\leq$ 641	173312.14
GEOM120b		21	189	$\leq$ 202	172800.01	$\leq$ 247	172852.82

Chakraborty, G. (2001). An Efficient Heuristic Algorithm for Channel Assignment Problem in Cellular Radio Networks. *IEEE Transactions on Vehicular Technology*, 50(6):1528–1539.

Costa, D. (1999). On the use of some known methods for T-coloring of graphs. *Annals of Operations Research*, 41:343–358.

Dias, B. (2014). Theoretical models and algorithms for optimizing channel assignment in wireless

Table 4: Results for the constraint and integer programming formulations applied on literature MS-FAP instances (Philadelphia, Helsinki and Artificial).

Const. matr.	Demd. vect.	Num. Vert.	Lower Bound	Chakraborty (2001)	Kendall and Mohamad (2005)	Kim et al. (2007)	Constr. Progr. (CP Optimizer)		Integer Progr. (CPLEX - B&C)	
				Best Reported	Best Reported	Best Reported	Best Found	Time (sec)	Best Found	Time (sec)
C_{21}^1	D_{21}^1	21	533	533	533	533	533	4.20	533	0.50
C_{21}^1	D_{21}^2		309	309	309	309	309	1.34	309	1.22
C_{21}^2	D_{21}^1		533	533	533	533	533	10.53	533	308.04
C_{21}^2	D_{21}^2		309	309	309	309	309	625.93	309	165.54
C_{21}^3	D_{21}^1		457	457	457	–	457	3.96	457	0.39
C_{21}^3	D_{21}^2		265	265	265	–	265	3.54	265	1.52
C_{21}^4	D_{21}^1		457	457	457	–	457	41.24	457	202.01
C_{21}^4	D_{21}^2		265	265	265	–	≤ 266	172800.06	265	214.01
C_{21}^5	D_{21}^1		381	381	381	381	381	3.23	381	0.29
C_{21}^5	D_{21}^2		221	221	221	221	221	100.81	221	5.09
C_{21}^6	D_{21}^1		381	463	435	427	≤ 449	172800.08	427	6827.49
C_{21}^6	D_{21}^2		221	273	268	253	≤ 266	172800.04	253	2026.67
C_{21}^7	D_{21}^1		305	305	305	–	305	12.85	305	1.10
C_{21}^7	D_{21}^2		177	197	185	–	≤ 180	172800.06	180	24.54
C_{21}^8	D_{21}^1		305	465	444	–	≤ 435	172800.07	427	1185.27
C_{21}^8	D_{21}^2		177	278	271	–	≤ 267	172800.06	253	1116.45
C_{25}^1	D_{25}^3	25	21	73	73	–	≤ 73	172800.00	73	1.10
C_{25}^1	D_{25}^4		89	121*	200	–	≤ 200	172800.07	200	2.18
C_{55}^1	D_{55}^5	55	309	309	309	–	309	11078.95	309	460.12
C_{55}^1	D_{55}^6		71	79	72	–	71	6.33	71	28.56

*Results lower than the obtained optimum - possibly wrong in the corresponding work.

mobile networks. Master's thesis, Federal University of Amazonas, Brazil. In portuguese.

Dias, B., Freitas, R., and Maculan, N. (2013). *Simulated annealing* para a alocação de canais em redes móveis celulares. In *Proceedings of the XLV Brazilian Symposium on Operations Research*. Sociedade Brasileira de Pesquisa Operacional (SOBRAPO). In portuguese.

Dorne, R. and Hao, J. (1998). Tabu search for graph coloring, t-coloring and set t-colorings. In Voss, S., Martello, S., Osman, I., and Roucairol, C., editors, *Metaheuristics: advances and trends in local search paradigms for optimization*, p. 77–92. Kluver Academic Publishers.

Galinier, P. and Hertz, A. (2006). A survey of local search methods for graph coloring. *Computers & Operations Research*, 33:2547–2562.

Gokbayrak, K. and Yıldırım, E. A. (2013). Joint gateway selection, transmission slot assignment, routing and power control for wireless mesh networks. *Computers & Operations Research*, 40 (7):1671–1679.

Hale, W. (1980). Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 25: 1497–1514.

Karp, R. (1972). Reducibility Among Combinatorial Problems. In Miller, R. E. and Thatcher, J. W., editors, *Complexity of Computer Computations*, p. 85–103. Plenum Press.

Kendall, G. and Mohamad, M. (2005). Solving the Fixed Channel Assignment Problem in Cellular Communications Using an Adaptive Local Search. In Burke, E. and Trick, M., editors, *5th International Conference for the Practice and Theory of Automated Timetabling (PATAT 2004)*, Lecture Notes in Computer Science, vol. 3616. Springer, Heidelberg.

Kim, S.-S., Smith, A. E., and Lee, J.-H. (2007). A memetic algorithm for channel assignment in wireless {FDMA} systems. *Computers & Operations Research*, 34(6):1842 – 1856.

Koster, A. (1999). *Frequency assignment: models and algorithms*. PhD thesis, Universiteit Maastricht, the Netherlands.

Lai, X. and Lü, Z. (2013). Multistart Iterated Tabu Search for Bandwidth Coloring Problem. *Computers & Operations Research*, 40:1401–1409.

Mak, V. (2007). Polyhedral studies for minimum-span graph labelling with integer distance constraints. *International Transactions in Operational Research*, 14(2):105–121.

Malaguti, E. and Toth, P. (2008). An evolutionary approach for bandwidth multicoloring problems. *European Journal of Operational Research*, 189(3):638–651.

Malaguti, E. and Toth, P. (2010). A survey on vertex coloring problems. *International Transactions in Operational Research*, 17(1):1–34.

Phan, V. and Skiena, S. (2002). Coloring graphs with a general heuristic search engine. In *Computational Symposium on Graph Coloring and Its Generalizations*, p. 92–99.

Trick, M., Mehrotra, A., and Johnson, D. (2002). COLOR02/03/04: Graph Coloring and its Generalizations. `http://mat.gsia.cmu.edu/COLOR02/`.