EURO Journal on Computational Optimization 10 (2022) 100049

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

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

Upper and lower bounds based on linear programming for the b-coloring problem

Roberto Montemanni [a,*], Xiaochen Chou [a], Derek H. Smith [b]

[a] *Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy*
[b] *School of Computing and Mathematics, University of South Wales, Llantwit Rd, Pontypridd, CF37 1DL, Wales, United Kingdom*

A R T I C L E I N F O

Keywords:
B-coloring
Linear programming
Graph coloring bounds

A B S T R A C T

B-coloring is a problem in graph theory. It can model some real applications, as well as being used to enhance solution methods for the classical graph coloring problem. In turn, improved solutions for the classical coloring problem would impact a larger pool of practical applications in several different fields such as scheduling, timetabling and telecommunications. Given a graph $G = (V, E)$, the *b-coloring problem* aims to maximize the number of colors used while assigning a color to every vertex in V, preventing adjacent vertices from receiving the same color, with every color represented by a special vertex, called a b-vertex. A vertex can be a *b-vertex* only if the set of colors assigned to its adjacent vertices includes all the colors, apart from the one assigned to the vertex itself.

This work employs methods based on Linear Programming to derive new upper and lower bounds for the problem. In particular, starting from a Mixed Integer Linear Programming model recently presented, upper bounds are obtained through partial linear relaxations of this model, while lower bounds are derived by considering different variations of the original model, modified to target a specific number of colors provided as input. The experimental campaign documented in the

* Corresponding author.
E-mail addresses: roberto.montemanni@unimore.it (R. Montemanni), xiaochen.chou@unimore.it (X. Chou), derek.smith@southwales.ac.uk (D.H. Smith).

https://doi.org/10.1016/j.ejco.2022.100049

paper led to several improvements to the state-of-the-art results.

1. Introduction

Given an undirected graph $G = (V, E)$, a b-coloring with K colors can be defined as a function that assigns a color $c(i) \in C = \{1, 2, \ldots, K\}$ to each vertex i of V, so that $c(i) \neq c(j)$ for every $(i, j) \in E$. Let $N(i) = \{j|(i, j) \in E\}$ be the neighborhood of i. For each $k \in C$ there must exist a vertex $i \in V$ with $c(i) = k$ and with $N(i) \cap \{j \in V|c(j) = h\} \neq \emptyset \; \forall h \in C \backslash \{k\}$. Less formally, it is required that for each color k used, there is a vertex assigned to color k (called *b-vertex*) such that for every other color used h, there is at least one of its neighbors assigned to h. A coloring of G with the minimum number $\chi(G)$ of colors must be a b-coloring. Otherwise each vertex assigned a color k which does not have a b-vertex could be re-colored with one of the colors other than k. This would contradict the minimality of $\chi(G)$.

The b-coloring problem aims to find a b-coloring using the maximum possible number of colors. Let $X_b(G)$ be the *b-chromatic number* of a graph G, defined as the maximum number of colors for which G admits a b-coloring. Fig. 1 provides an example of an optimal b-coloring.

2. Literature review

Estimating $X_b(G)$ is proved to be NP-hard in [11]. Consequently, the b-coloring problem is also NP-hard. It has been proved in [15] that the difference between the optimal solution values of the classical coloring problem ([16]) and b-coloring for the same graph G can be arbitrarily large. The b-coloring problem can be largely influenced by the girth (length of a shortest cycle) of the graph, as shown in [3]. As demonstrated in [1], a b-coloring with k colors does not necessarily exist for all the possible values of k ranging from the minimum number for which a b-coloring exists up to the b-chromatic number; gaps might exist.

A hybrid evolutionary algorithm for the b-coloring problem is discussed in [6]. A integer linear programming formulation for the b-chromatic index $X_b(G)$ is introduced in [14], and this model is at the basis of the branch and cut algorithm provided in [13]. Matheuristic approaches to the problem are introduced in [17]. In the same work, an effective new mixed integer linear programming model is presented. Another metaheuristic method, based on an iterative schema and able to improve some of the lower bounds on the same set of instances, was discussed in [18]. Note that the testbed commonly adopted

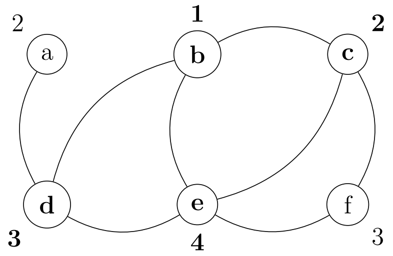

Fig. 1. Example of a graph with an associated optimal b-coloring with 4 colors, here represented by numbers. Vertices with the name in bold are the b-vertices.

for b-coloring is composed of instances originally proposed for other graph problems in [12].

In the works [7] and [8] b-coloring is used within postal mail sorting systems to model a new approach for address block localization. The aim is to assist the software for address recognition. A novel clustering technique based on b-coloring is used by the French healthcare system to identify and formalize a new typology of hospital stays, as presented in [5]. As discussed in [2], an important indirect practical motivation for attacking the b-coloring problem is that it can provide viable bounds for the classical coloring problem. Note that this in turn may potentially lead to benefits to several important practical applications such as scheduling [21], timetabling [4] and telecommunications [19,20,9].

The paper is organized as follows. A Mixed Integer Program is discussed in Section 3. This model will be the starting point for the subsequent results. In Section 4 some upper bounding technique based on partial linear relaxations are proposed. Section 5 is devoted to heuristic solutions and lower bounds, while Section 6 presents and summarizes the computational results of the methods previously described. Section 7 finally concludes the paper.

3. An integer programming model

In this section an Integer Programming model originally proposed in [17] is described. There is a set of variables x such that $x_{ij} = 1$ if vertex j is colored with the color of the representative vertex i, 0 otherwise. A vertex i is a b-vertex, or representative, if and only if $x_{ii} = 1$. In order to simplify the notation, let $\bar{N}(i) = V \setminus \{\{i\} \cup N(i)\}$ be the anti-neighborhood of i.

$$BC: \quad \max \quad \sum_{i \in V} x_{ii} \tag{1}$$

$$x_{ij} + x_{ik} \leq x_{ii} \qquad i \in V; j,k \in \bar{N}(i); (j,k) \in E \tag{2}$$

$$x_{ij} \leq x_{ii} \qquad i \in V; j \in \bar{N}(i); \nexists k \in \bar{N}(i) : (j,k) \in E \tag{3}$$

$$\sum_{\substack{k \in N(j), \\ k \in \bar{N}(i)}} x_{ik} \geq x_{ii} + x_{jj} - 1 \qquad i,j \in V; (i,j) \notin E \tag{4}$$

$$\sum_{j \notin N(i)} x_{ji} = 1 \qquad i \in V \qquad (5)$$

$$x_{ij} \in \{0, 1\} \qquad i, j \in V \qquad (6)$$

The objective function (1) aims at maximizing the number of b-vertices. Constraints (2) imposes a proper coloring, and at the same time allows a vertex to give a color only if it is a representative. Constraints (3) again state that only representative can give a color, and applies to those cases that are not already covered by (2). Constraints (4) formalize the proper b-coloring restrictions. They imply that if both vertices i and j are b-vertices, then there must be a neighbor of j which is represented by i. Technically, if both i and j are representatives then the right-hand side is equal to one, implying that the summation in the left-hand side (composed by the neighbors of j potentially represented by i) should be at least one. Constraints (5) ensure that every vertex must be assigned a color (note that j can take value i in the summation). The domain definition for the variables is provided by constraints (6). We refer the interested reader to [17] for a more in-depth discussion of the model.

4. Upper bounds based on partial linear relaxations

The linear relaxation of model (BC) is obtained by substituting constraints (6) with the following ones:

$$0 \leq x_{ij} \leq 1 \quad i, j \in V \qquad (7)$$

By definition, the cost of the optimal solution of the linear relaxation provides a valid upper bound for the optimal cost of BC, the domains of the variables defined by (7) being a relaxation of the domains originally specified by (6). The linear relaxation is also much easier to solve with respect to the original BC, which is an integer program. As a consequence, solving the linear relaxation could turn out to be a suitable method to efficiently derive effective upper bounds for the costs of b-coloring problems.

Solving BC provides optimal solutions but often has impractical computation times, while solving the pure linear relaxation provides (possibly weak) upper bounds with a low computational effort. It is also possible to consider intermediate versions, namely partial linear relaxations, where only a fraction of the variables are forced to be integer through constraint (6), while the remaining variables are continuous, as in constraint (7). The previous studies [17] and [18] indicated that the representative-selection variables x_{ii} are the critical ones, since once they are set, the complexity of the residual problem boils down substantially. For this reason, our strategy will only operate on them, with the remaining x_{ij} variables continuous. In particular, we will refer to these models as BC(p), where p indicates the percentage of the representative variables x_{ii} forced to be integer. With such a notation, BC(100) represents a mixed integer program, while BC(0) is the pure linear relaxation. Note that even when large values of p are considered, only a fraction of the total variables are set to binary, since only representative variables are considered while calculating the percentage p. Anyway, all the partial linear relaxations considered for a same instance can be seen as a tradeoff between precision and speed, as it will be shown in Section 6.2.1.

When a partial linear relaxation approach is considered, a further question to answer is about the selection of the $p\%$ of the representative variables for which integrality should be enforced. Similar questions have been shown to be relevant for related problems such as the maximum clique problem [22]. In our case, preliminary results have clearly shown that using heuristic criteria, taking into account vertex characteristics such as the number of neighbors, does not lead to any advantage. In the rest of the paper the $p\%$ of variables forced to be integer are therefore selected at random. This means that different runs might lead to different results. In our experiments we will consider one run only. Note that more runs could have produced better results and give more clues about the robustness of the methods, but the computational power available to us was limited. The tests proposed however clearly show the potential of the methods proposed.

5. Lower bounds based on models for the decision version of the problem

5.1. Model $BC^x(T)$

Formally, after having decided a target value T for the objective function (1), starting from model BC it is possible to obtain the following decision model $BC^x(T)$.

$$BC^x(T): \quad x_{ij} + x_{ik} \leq x_{ii} \qquad i \in V; j,k \in \bar{N}(i); (j,k) \in E \tag{8}$$

$$x_{ij} \leq x_{ii} \qquad i \in V; j \in \bar{N}(i); \nexists k \in \bar{N}(i) : (j,k) \in E \tag{9}$$

$$\sum_{\substack{k \in N(j),\\ k \in \bar{N}(i)}} x_{ik} \geq x_{ii} + x_{jj} - 1 \qquad i,j \in V; (i,j) \notin E \tag{10}$$

$$\sum_{j \notin N(i)} x_{ji} = 1 \qquad i \in V \tag{11}$$

$$\sum_{i \in V} x_{ii} = T \tag{12}$$

$$x_{ij} \in \{0,1\} \qquad i,j \in V \tag{13}$$

The new model neglects the objective function (1) and has the new constraint (12) to define the target number of colors. The constraint (12) implies that we accept only feasible solutions with exactly T b-vertices.

Note that if a feasible solution with T b-vertices exists for an instance, then the model will return a feasible solution. Otherwise, the solver will return an appropriate message to signal an infeasible problem, and it is possible to conclude that no feasible solution exists with the given number T of b-vertices.

The model $BC^x(T)$ is a decision model; if a feasible solution is identified, the computation is stopped, or alternatively the computation stops once the solver proves that such a solution does not exist. It is possible that faster feasible colorings with T colors can be obtained by models with a cost function. For these models solutions with a non-zero cost may exist, although only solutions with cost zero correspond to feasible colorings. Having a path of solutions with decreasing costs toward a feasible (zero cost) solution might help the solver. This is the rationale behind the models described in the following sections.

5.2. Model $BC^y(T)$

Starting from the base model $BC^x(T)$, for this model a new set of non-negative integer variables y defined for each $i \in V$ are introduced such that y_i represents the number of excess colors assigned to vertex i, additional to the single color required. With this new set of variables, it is possible to define the following new model $BC^y(T)$ for the decision version of b-coloring. Unchanged constraints with respect to $BC^x(T)$ are repeated for the sake of clarity.

$$BC^y(T): \quad \min \sum_{i \in V} y_i \tag{14}$$

$$x_{ij} + x_{ik} \leq x_{ii} \qquad i \in V; j,k \in \bar{N}(i); (j,k) \in E \tag{15}$$

$$x_{ij} \leq x_{ii} \qquad i \in V; j \in \bar{N}(i); \nexists k \in \bar{N}(i) : (j,k) \in E \tag{16}$$

$$\sum_{\substack{k \in N(j), \\ k \in \bar{N}(i)}} x_{ik} \geq x_{ii} + x_{jj} - 1 \qquad i,j \in V; (i,j) \notin E \tag{17}$$

$$\sum_{j \notin N(i)} x_{ji} = 1 + y_i \qquad i \in V \tag{18}$$

$$\sum_{i \in V} x_{ii} = T \tag{19}$$

$$x_{ij} \in \{0,1\} \qquad i,j \in V \tag{20}$$

$$y_i \geq 0, \text{ integer} \qquad i \in V \tag{21}$$

A feasible solution with T colors for the b-coloring problem exists if and only if a solution of $BC^y(T)$ of cost 0 exists, according to the new objective function (14). Note that now feasible solutions with non-zero cost for the model are allowed, and the hope is that this can give an advantage to the solver. The other differences with respect to the standard model $BC^x(T)$ are constraints (18) that substitute (11) and the presence of the domain constraints (21) for the y_i variables.

5.3. Model $BC^z(T)$

Starting from the base model $BC^x(T)$, for this model a new set of binary variables z defined for each $(i,j) \in E$ are introduced such that $z_{ij} = 1$ if nodes i and j are assigned the same color notwithstanding $(i,j) \in E$, 0 otherwise. With this new set of variables, it is possible to define the following new model $BC^z(T)$ for the decision version of b-coloring. Unchanged constraints with respect to $BC^x(T)$ are repeated for the sake of clarity.

$$BC^z(T): \quad \min \sum_{i \in V} \sum_{j \in V:(i,j) \in E} z_{ij} \tag{22}$$

$$x_{ij} + x_{ik} \le x_{ii} + z_{jk} \qquad i \in V; j,k \in \bar{N}(i); (j,k) \in E \tag{23}$$

$$x_{ij} \le x_{ii} \qquad i \in V; j \in \bar{N}(i); \nexists k \in \bar{N}(i) : (j,k) \in E \tag{24}$$

$$\sum_{\substack{k \in N(j), \\ k \in \bar{N}(i)}} x_{ik} \ge x_{ii} + x_{jj} - 1 \qquad i,j \in V; (i,j) \notin E \tag{25}$$

$$\sum_{j \notin N(i)} x_{ji} = 1 \qquad i \in V \tag{26}$$

$$\sum_{i \in V} x_{ii} = T \tag{27}$$

$$x_{ij} \in \{0,1\} \qquad i,j \in V \tag{28}$$

$$z_{ij} \in \{0,1\} \qquad (i,j) \in E \tag{29}$$

A feasible solution with T colors for the b-coloring problem exists if and only if a solution of $BC^z(T)$ of cost 0 exists, according to the new objective function (22). Note that now feasible solutions with non-zero cost for the model are allowed, and, as explained previously, the hope is that this can give an advantage to the solver. The other differences with respect to the standard model $BC^x(T)$ are constraints (23) that substitute (8) and the presence of the domain constraints (29) for the z_{ij} variables.

5.4. Model $BC^w(T)$

Starting from the base model $BC^x(T)$, for this model a new set of binary variables w defined for each $(i,j) \notin E$ are introduced such that $w_{ij} = 1$ if both vertices i and j are representative but there is no vertex adjacent to j associated with the color of i (violating therefore a crucial property of b-coloring), 0 otherwise. With this new set of variables, it is possible to define the following new model $BC^w(T)$ for the decision version of b-coloring. Unchanged constraints with respect to $BC^x(T)$ are repeated for the sake of clarity.

$$BC^w(T): \quad \min \sum_{i\in V} \sum_{j\in V:(i,j)\notin E} w_{ij} \tag{30}$$

$$x_{ij} + x_{ik} \le x_{ii} \qquad i \in V; j,k \in \bar{N}(i); (j,k) \in E \tag{31}$$

$$x_{ij} \le x_{ii} \qquad i \in V; j \in \bar{N}(i); \nexists k \in \bar{N}(i) : (j,k) \in E \tag{32}$$

$$\sum_{\substack{k\in N(j),\\ k\in \bar{N}(i)}} x_{ik} \ge x_{ii} + x_{jj} - 1 - w_{ij} \qquad i,j \in V; (i,j) \notin E \tag{33}$$

$$\sum_{j\notin N(i)} x_{ji} = 1 \qquad i \in V \tag{34}$$

$$\sum_{i\in V} x_{ii} = T \tag{35}$$

$$x_{ij} \in \{0,1\} \qquad i,j \in V \tag{36}$$

$$w_{ij} \in \{0,1\} \qquad (i,j) \notin E \tag{37}$$

A feasible solution with T colors for the b-coloring problem exists if and only if a solution of $BC^w(T)$ of cost 0 exists, according to the new objective function (30). Note that now feasible solutions with non-zero cost for the model are allowed, and, as explained previously, the hope is that this can give an advantage to the solver. The other differences with respect to the standard model $BC^x(T)$ are constraints (33) that substitute (10) and the presence of the domain constraints (37) for the w_{ij} variables.

6. Experimental results

6.1. Datasets and settings

The instances considered for the experiments of this paper are based on the DIMACS benchmark set originally proposed for the minimum coloring and the maximum clique problems in [12]. The instances have been considered for the first time in the b-coloring context in [17], where a total of 59 instances from minimum coloring and 78 from maximum clique have been adopted. Due to the nature of Linear Programming and the characteristics of the models, it is not possible to handle all the instances (typically those with more than 500 nodes are out of reach), moreover the instances for which an optimal solution is already known have not been considered in this study. This leaves us roughly with a total of 32 instances from minimum coloring and 55 instances from maximum clique, although not all the methods proposed will be able to handle all of the instances.

The routines to create all the models considered have been coded in ANSI C, and all the experiments reported have been run on a computer equipped with an Intel Core i7 processor running at 2.7 GHz and 16 GB of RAM running Windows 10. All the linear and integer models have been solved by Gurobi 9.1 [10] running in single-thread mode.

In all the experiments reported a maximum computation time of 1800 seconds is allowed for each method/instance combination.

6.2. Partial linear relaxations

6.2.1. Percentage of integer variables

The aim of this section is to investigate the trade-off between quality of the upper bound and computation time while using the partial linear relaxation approach described in Section 4. We consider three representative instances and chart the values of the solution of $BC(p)$ and the relative computation time for several values of the percentage p of representative variables set to be integer. The results are depicted in Fig. 2. One can observe how in general the quality of the upper bound gets better when higher values of p are considered, although the computation time required to produce the estimation increases very quickly with p. This is especially true for the instance *R2501c*, that does not appear particularly challenging for the solver: all the partial linear relaxations are solved to optimality. For the other two instances, when the value of p is increased, the computation is interrupted after 1800 seconds, and the value of the solution reported is that of the best heuristic solution found at that time, which is probably not optimal. The lack of optimality is clear for the instance *gen200_p0.9_55*, for which the upper bound oscillates.

In conclusion, considering partial linear relaxation with an increasing percentage of integer variables moderately improves the quality of the upper bound, at the price of a substantial increase in the computation time. The technique looks, however, to be effective.

6.2.2. Results

In this section we summarise the results obtained by $BC(p)$ for different values of p, namely $0, 5, 10, 20, 40, 60, 80, 100$. Only those instances for which a solution was obtained by at least one of the values of p considered within 1800 seconds, are included in the tables. The columns of Tables 1 and 2 contain, for each instance reported, the following information:

- Name: name of the instance;
- V: number of vertices of the graph;
- E: number of edges of the graph;
- Best known UB: the best-known upper bound for the value of $X_b(G)$ available before the present study (from [17]);
- Best $BC(p)$: the best upper bound retrieved by $BC(p)$ and the value of p corresponding to the best (in case of ties the value of p leading to the fastest solution is reported). Improved upper bounds are reported in bold font.

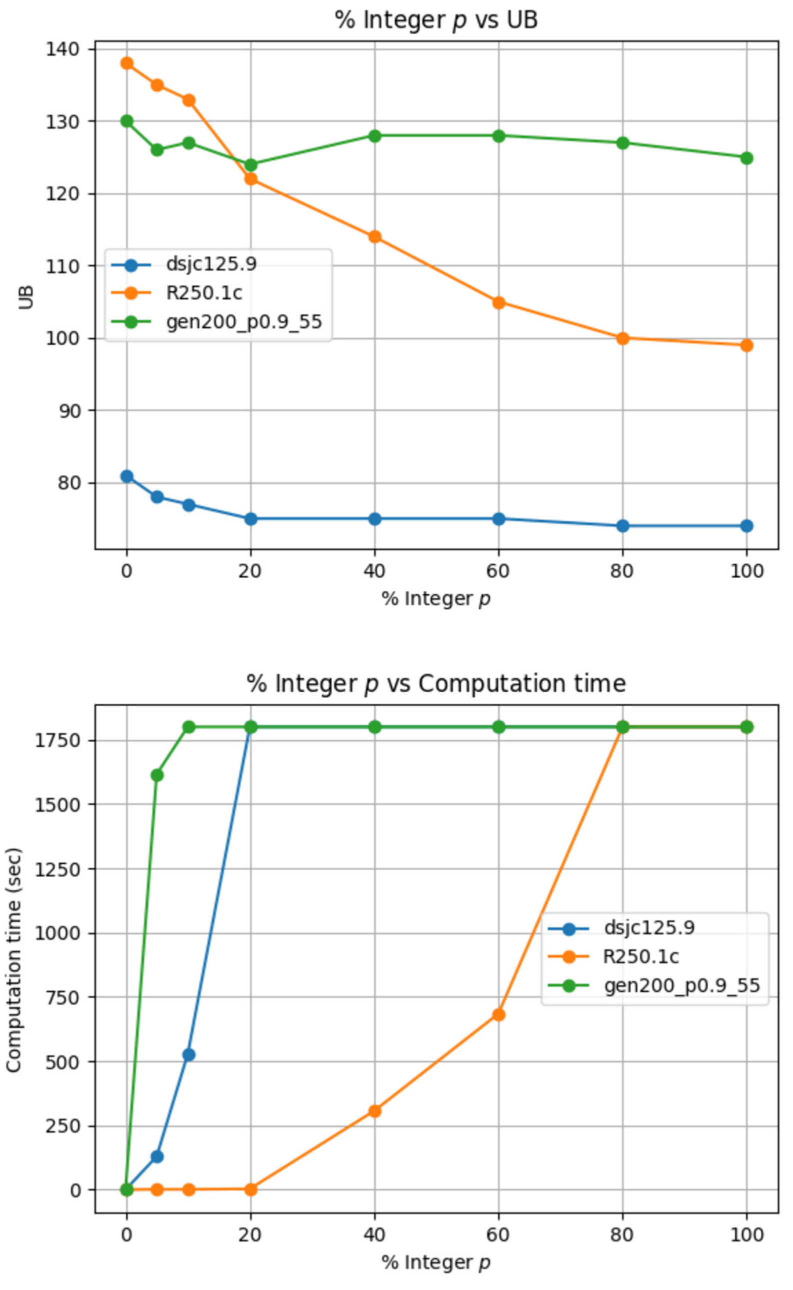

Fig. 2. Evolution of the upper bounds and computation times while varying the percentage of integer variables p in $BP(p)$.

Table 1 covers 44 instances, and 38 improved upper bounds are reported. This indicates that working on partial linear relaxations is indeed a promising approach. Looking at the values of p leading to the best results within the 1800 seconds allowed, one can

Table 1
Best upper bounds retrieved by $BC(p)$ for clique instances from [12]. Improved upper bounds are shown for 38 of the 44 instances considered.

Instance			Best known UB [17]	Best $BC(p)$	
Name	$\|V\|$	$\|E\|$		UB	% integer p
brock200_1	200	14834	146	**127**	0
brock200_2	200	9876	100	119	0
brock200_3	200	12048	120	123	0
brock200_4	200	13089	129	**124**	0
brock400_1	400	59723	294	**254**	0
brock400_2	400	59786	295	**254**	0
brock400_3	400	59681	294	**254**	5
brock400_4	400	59765	295	**254**	0
C125.9	125	6963	108	**74**	40
C250.9	250	27984	220	**162**	5
C500.9	500	112332	442	**327**	0
gen200_p0.9_44	200	17910	174	**123**	40
gen200_p0.9_55	200	17910	174	**124**	20
gen400_p0.9_55	400	71820	348	**261**	0
gen400_p0.9_65	400	71820	350	**262**	0
gen400_p0.9_75	400	71820	350	**262**	0
hamming6-2	64	1824	58	**36**	60
hamming6-4	64	704	23	**22**	60
hamming8-2	256	31616	248	**161**	10
hamming8-4	256	29864	164	**144**	5
johnson8-2-4	28	210	16	**11**	60
johnson8-4-4	70	1855	54	**36**	40
johnson16-2-4	120	5460	92	**55**	60
johnson32-2-4	496	107880	436	**262**	20
keller4	171	9435	106	**101**	0
MANN_a9	45	918	41	**21**	60
MANN_a27	378	70551	365	**149**	80
p_hat300-1	300	10933	91	300	0
p_hat300-2	300	21928	149	177	0
p_hat300-3	300	33390	209	**190**	5
san200_0.7_1	200	13930	138	**126**	40
san200_0.7_2	200	13930	134	**116**	5
san200_0.9_1	200	17910	173	**112**	60
san200_0.9_2	200	17910	175	**124**	80
san200_0.9_3	200	17910	176	**124**	60
san400_0.5_1	400	39900	204	220	0
san400_0.7_1	400	55860	277	**253**	0
san400_0.7_2	400	55860	277	**251**	0
san400_0.7_3	400	55860	274	**248**	0
san400_0.9_1	400	71820	353	**264**	0
sanr200_0.7	200	13868	137	**125**	0
sanr200_0.9	200	17863	175	**127**	5
sanr400_0.5	400	39984	201	400	0
sanr400_0.7	400	55869	276	**251**	5

observe an inverse correlation between the number of nodes $|V|$ and edges $|E|$ and the best value of p. This suggests that the size of the instances is an indicator about how challenging the instances are for the solver. There are however outliers like, for example, *MANN_a27*, which is a fairly large instance, but for which the best result is obtained for $p = 80$. This happens because a good heuristic solution (upper bound) is retrieved by the solver within the allowed time. In conclusion, a forecast for the best value of p can be done by considering the size of the instance under investigation, but this factor does not fully capture the essence of the problem, and can only be used as a rough indicator.

Table 2
Best upper bounds retrieved by $BC(p)$ for coloring instances from [12]. Improved upper bounds are shown for 5 of the 15 instances considered.

Instance			Best known UB [17]	Best $BC(p)$	
Name	$\|V\|$	$\|E\|$		UB	% integer p
dsjc125.5	125	3891	63	75	0
dsjc125.9	125	6961	109	**74**	40
dsjc250.5	250	15668	126	150	0
dsjc250.9	250	27897	219	**162**	5
dsjr500.1c	500	121275	478	**228**	80
flat300_26_0	300	21633	146	179	0
fpsol2.i.1	451	8691	79	151	0
mulsol.i.1	197	3925	65	67	40
mulsol.i.2	188	3885	53	69	40
r125.1	125	209	7	38	40
r125.1c	125	7501	116	**54**	80
r250.1c	250	30227	238	**99**	100
r250.5	250	14849	119	150	0
school1_nsh	352	14612	101	351	0
school1	385	19095	117	384	0

Similar conclusions can be drawn for the results reported in Table 2, although in this case only 5 upper bounds have been improved over the 15 instances considered. Note also that for this second set of instances the correlation between the size of the instances and the best value of p appears to be less obvious, making the prediction of the right value of p more difficult.

Observe finally how for two instances – p_hat300-1 and sanr400_0.5 in Table 1 – only a trivial upper bound equal to the number of nodes is reported, indicating that the upper bounding methods proposed are not able to properly handle these instances. The heuristic methods we propose will be however able to improve the best known lower bounds for these instances (see Section 6.3.2)..

6.3. Models for the decision version of b-coloring

6.3.1. Comparison of the models

In this section we consider some representative instances and after having set a value of T equal to the cost of the best-known heuristic solution, we report the results obtained by the four models described in Section 5. Note that in the experiments we purposely run some values of T equal to the known optimal solution plus one, in order to test the different models on infeasible problems. The results are reported only for instances for which at least one of the models considered was able to return a conclusive answer.

The columns of Tables 3 and 4 contain, for each instance reported, the following information:

- Name: name of the instance;
- V: number of vertices of the graph;
- E: number of edges of the graph;

Table 3
Lower bounds computation results on the relevant clique instances from [12].

Instance			T	$BC^x(T)$		$BC^y(T)$		$BC^z(T)$		$BC^w(T)$	
Name	$\|V\|$	$\|E\|$		Res	Sec	Res	Sec	Res	Sec	Res	Sec
c-fat200-1	200	1534	19	N	120	N	280	N	182	-	-
c-fat200-2	200	3235	35	N	260	-	-	N	874	-	-
c-fat200-5	200	8473	87	N	699	N	1072	-		-	-
c-fat500-1	500	4459	22	N	1394	-	-	N	1604	-	-
C250.9	250	27984	128	Y	241	Y	1063	-		-	-
gen200_p0.9_55	200	17910	105	-	-	-	-	-		Y	1510
hamming6-2	64	1824	36	-	-	N	202	N	1166	N	95
hamming8-4	256	20864	53	Y	13	Y	1139	-	-	-	-
johnson8-2-4	28	210	10	N	3	N	33	N	8	N	26
johnson32-2-4	496	107880	41	-	-	Y	246	-		-	
MANN_a9	45	918	22	N	1	N	3	N	0	N	0
MANN_a27	378	70551	145	N	10	N	270	N	181	N	127
p_hat300-1	300	10933	48	-	-	-	-	-	-	Y	501
p_hat300-2	300	21928	83	-	-	Y	453	-	-	-	-
p_hat500-3	500	93800	152	Y	1053	Y	150	-	-	-	-
san200_0.9_2	200	17910	106	-	-	-	-	-	-	Y	1249
san200_0.9_3	200	17910	104	Y	150	Y	349	Y	182	Y	1206
san400_0.7_3	400	55860	81	-	-	Y	369	Y	799	-	-
sanr200_0.7	200	13868	68	-	-	Y	1431	-	-	-	-
sanr200_0.9	200	17863	104	Y	199	-	-	-	-	Y	628
sanr400_0.5	400	39984	75	-	-	-	-	-	-	Y	419
sanr400_0.7	400	55869	106	Y	1320	-	-	-	-	-	-

Table 4
Lower bounds computation results on the relevant coloring instances from [12].

Instance			T	$BC^x(T)$		$BC^y(T)$		$BC^z(T)$		$BC^w(T)$	
Name	$\|V\|$	$\|E\|$		Res	Sec	Res	Sec	Res	Sec	Res	Sec
dsjc125.1	125	736	18	N	115	-	-	N	139	-	-
dsjc250.9	250	27897	128	Y	861	-	-	-	-	Y	1109
DSJR500.1c	500	121275	143	-	-	Y	649	Y	258	Y	241
inithx.i.1	864	18707	78	Y	2	-	-	-	-	-	-
le450_5c	450	9803	36	-	-	-	-	-	-	Y	472
le450_15c	450	16680	55	Y	475	-	-	-	-	-	-
R125.1	125	209	8	N	22	N	235	N	18	-	-
R125.1c	125	7501	54	N	0	N	1	N	1	N	0
R250.1	250	867	13	N	1222	N	1410	-	-	-	-
zeroin.i.1	211	4100	55	N	1165	N	178	-	-	N	1707
zeroin.i.2	211	3541	42	N	1534	-	-	-	-	-	-

- T: target value for the number of colors;
- For each different model $BC^*(T)$ considered we report:
  - Res: the outcome of the solver on the model: “Y” means a feasible b-coloring with T colors has been retrieved; “N” means it has been proved that a b-coloring with T colors does not exist; “-” means that the computation has been inconclusive within the 1800 seconds allowed;
  - Sec: the total computation time (approximated to seconds) taken to solve the model. In case of an inconclusive run, “-” is reported.

From Tables 3 and 4 the base model $BC^x(T)$ appears to be the most effective and fastest method in general. This was expected, since it has fewer variables and constraints than the other models. On the other hand, there are also several instances for which the other models are able to return an answer while $BC^x(T)$ fails. These are the improvements we were hoping for. In particular, models $BC^y(T)$ and $BC^w(T)$ appear to be able to succeed in finding a feasible solution (answer "Y") for some instances while all the other methods are unable to close the computation within the 1800 seconds allowed. On the other hand, model $BC^z(T)$ appears to be always dominated on the instances considered (it is the fastest one reporting an answer only for instance *R125.1*), but it is interesting to observe that with respect to the other methods, it seems consistently effective in identifying infeasible models (answers "N"). The results justify a full experimental campaign, where all the four methods are run on all the instances in reach, with increasing values of T (starting from the best-known solution plus one) as far as possible.

6.3.2. New heuristic solutions

The models presented in Section 5 can be used to retrieve improved heuristic solutions, by setting the value of T to values higher than the currently best known lower bound. We systematically run the four models discussed in Section 5 on each instance, starting from a value of T equal to the current best known result from [17] and [18], increased by one. Given each instance and each model, the value of T was repeatedly increased by one until the instance was not closed in the given 1800 seconds. The best value of T for which a solution had been retrieved (answer "Y") is then reported as the best improved heuristic solution retrieved.

The columns of Tables 5 and 6 contain, for each instance reported, the following information:

- Name: name of the instance;
- V: number of vertices of the graph;
- E: number of edges of the graph;
- Best known LB: the best-known lower bound for the value of $X_b(G)$ available before the present study (from [17] and [18]);
- New best $BC^*(T)$: the best lower bound retrieved by the models described in Section 5. The value (*LB*) and the model able to obtain it (*Model*) are reported.

From the results presented in Tables 5 and 6 it is possible to appreciate the contributions of models $BC^x(T)$, $BC^y(T)$ and $BC^w(T)$ to the new best heuristic solutions. This indicates that considering the models with more variables and constraints could be a good strategy to increase the probability of retrieving high quality solutions.

6.3.3. Upper bounds reduction

In addition to the natural approach to retrieve new heuristic solutions, the models discussed in Section 5 can be used in a reverse way to prove no solution exists for a given

Table 5
Improved heuristic solutions for the clique instances from [12].

Instance			Best known LB [17], [18]	New best $BC^*(T)$	
Name	$\|V\|$	$\|E\|$		LB	Model
C250.9	250	27984	127	128	x
gen200_p0.9_55	200	17910	104	105	w
hamming8-4	256	29864	48	56	x
johnson32-2-4	496	107880	40	42	y
p_hat300-1	300	10933	39	48	w
p_hat300-2	300	21928	71	85	y
p_hat500-3	500	93800	151	152	x
san200_0.9_2	200	17910	105	106	w
san200_0.9_3	200	17910	103	104	x
san400_0.7_3	400	55860	80	82	y
sanr200_0.7	200	13868	67	68	y
sanr200_0.9	200	17863	103	104	x
sanr400_0.5	400	39984	74	75	w
sanr400_0.7	400	55869	105	106	x

Table 6
Improved heuristic solutions for the coloring instances from [12].

Instance			Best known LB [17], [18]	New best $BC^*(T)$	
Name	$\|V\|$	$\|E\|$		LB	Model
dsjc250.9	250	27897	127	128	x
DSJR500.1c	500	121275	142	150	y
le450_5c	450	9803	35	36	w
le450_15c	450	16680	54	55	x

instance and a given value of T, with the aim of reducing known upper bounds. Starting from the current best-known upper bounds from [17] and Section 6.2, we systematically run the four models discussed in Section 5 on each instance. Given each instance and each model, the value of T was repeatedly decreased by one until the instance was not closed in the given 1800 seconds, or the best known lower bound (from [17], [18] and Section 6.3.3) was matched. In case during the process a solution should be provided (answer "Y"), it would be a new best, but this never happens in our experiments. Every run for which an answer "N" is returned, coincides with a value of T for which no solution exists. A chain of consecutive values starting from the best known upper bounds for which no solution exists, means an improved upper bound.

The columns of Tables 7 and 8 contain, for each instance reported, the following information:

- Name: name of the instance;
- V: number of vertices of the graph;
- E: number of edges of the graph;
- Best known UB: the best-known upper bound for the value of $X_b(G)$ from [17] and Section 6.2;

Table 7
Improved upper bounds from infeasible models for the clique instances from [12].

Instance			Best known UB [17], Section 6.2	New best $BC^*(T)$	
Name	$\|V\|$	$\|E\|$		UB	Model
C125.9	125	6963	72	71	z
C250.9	250	27984	162	152	x
gen200_p0.9_44	200	17910	123	118	x
gen200_p0.9_55	200	17910	124	119	x
hamming6-2	64	1824	36	35	w
hamming8-2	256	31616	161	160	x
johnson8-2-4	28	210	11	9	z
johnson8-4-4	70	1855	36	35	x
johnson16-2-4	120	5460	55	46	x
MANN_a27	378	70551	149	144	z
san200_0.7_1	200	13930	126	124	x
san200_0.7_2	200	13930	116	115	x
san200_0.9_1	200	17910	112	111	x
san200_0.9_2	200	17910	124	119	x
san200_0.9_3	200	17910	124	119	x
san400_0.9_1	400	71820	264	259	x
sanr200_0.9	200	17863	127	119	x
sanr400_0.5	400	39984	400	385	z

Table 8
Improved upper bounds from infeasible models for the coloring instances from [12].

Instance			Best known UB [17], Section 6.2	New best $BC^*(T)$	
Name	$\|V\|$	$\|E\|$		UB	Model
dsjc125.9	125	3891	74	72	x
dsjc250.9	250	27897	162	150	x
DSJR500.1c	500	121275	228	221	x
R125.1c	125	7501	54	53	x
R250.1c	250	30227	100	89	x

- New best $BC^*(T)$: the best upper bound retrieved by proving a model described in Section 5 is infeasible for some sequential values of T. The value (UB) and the model used to derive it ($Model$) are reported.

The results of Tables 7 and 8 suggest that models can be used to refine upper bounds, by proving some values of the number of colors T infeasible. Note that most of the improvements have been obtained with the standard model $BC^x(T)$, but also model $BC^z(T)$ substantially contributed by proving four new bounds.

6.4. Summary of the improved bounds

In this section we summarise all the improved upper and lower bounds found in our study. Only the relevant instances are included in the table. The columns of Tables 9 and 10 contain, for each instance reported, the following information:

- Name: name of the instance;
- V: number of vertices of the graph;

- E: number of edges of the graph;
- Lower Bounds: the best-known lower bound for the value of $X_b(G)$ available before the present study (from [17] and [18], *Old*) and the new bounds retrieved in this study (*New*);
- Upper Bounds: the best-known upper bound for the value of $X_b(G)$ available before the present study (from [17] and [18], *Old*) and the new bounds retrieved in this study (*New*);
- Newly Closed: This column contains an asterisk in correspondence of the instances for which optimality of a solution has been provided for the first time in this study.

The tests for lower bounds have been run on 55 instances out of the 78 of the testbed introduced in [17], leaving out those instances already closed, or out of reach for the current methods due to their size. Improved results are summarized in Table 9 (note that non relevant instances are omitted). According to the table, 13 improved heuristic solutions (lower bounds) have been retrieved for clique instances by the new methods proposed. The newly proposed upper bounding techniques were instead able to improve 38 best-known upper bounds. Finally, the optimality of the solution of 4 instances has been provided for the first time.

The tests for heuristic solutions have been run on 32 instances out of the 59 of the testbed introduced in [17], leaving out those instances already closed, or out of reach for the current methods due to their size. Improved results are summarized in Table 10 (note that non relevant instances are omitted). According to the table, 4 improved heuristic solutions (lower bounds) have been retrieved for coloring instances by the new methods proposed. The newly proposed upper bounding techniques were instead able to improve 11 best-known upper bounds. Finally, the optimality of the solution of 7 instances has been provided for the first time.

It is worth observing that for all the newly closed instances reported in Tables 9 and 10, the optimal lower bound was known already, and the optimality proof has been provided by reducing the upper bound. This can be read as an indication of the quality of the heuristic methods previously available, and of those introduced for the first time in the present study.

7. Conclusions

In this paper we presented several ideas on how to use Linear and Mixed-Integer Programming to obtain upper and lower bounds for the optimal solution cost of b-coloring, given an input graph. Such improvements have a direct potential impact on some real applications that can be modeled as b-coloring problems, and an indirect impact on the many applications that can be modeled with traditional graph coloring, since b-coloring can improve some bounding techniques for the classical coloring problem.

Table 9
Improved bounds for the clique instances from [12].

Instance			Lower Bounds		Upper Bounds		Newly Closed
Name	$\|V\|$	$\|E\|$	Old	New	Old	New	
brock200_1	200	14834	73		146	127	
brock200_4	200	13089	63		129	124	
brock400_1	400	59723	123		294	254	
brock400_2	400	59786	121		295	254	
brock400_3	400	59681	123		294	254	
brock400_4	400	59765	125		295	254	
C125.9	125	6963	68		108	71	
C250.9	250	27984	127		220	152	
C500.9	500	112332	250		442	327	
gen200_p0.9_44	200	17910	104		174	118	
gen200_p0.9_55	200	17910	104	105	174	119	
gen400_p0.9_55	400	71820	200		348	261	
gen400_p0.9_65	400	71820	200		350	262	
gen400_p0.9_75	400	71820	200		350	262	
hamming6-2	64	1824	35		58	35	*
hamming6-4	64	704	15		23	22	
hamming8-2	256	31616	144		248	160	
hamming8-4	256	29864	48	56	164	144	
johnson8-2-4	28	210	9		16	9	*
johnson8-4-4	70	1855	28		54	35	
johnson16-2-4	120	5460	37		92	46	
johnson32-2-4	496	107880	40	42	436	262	
keller4	171	9435	48		106	101	
MANN_a9	45	918	21		41	21	*
MANN_a27	378	70551	144		365	144	*
p_hat300-1	300	10933	39	48	91		
p_hat300-2	300	21928	71	85	149		
p_hat300-3	300	33390	113		209	190	
p_hat500-3	500	93800	151	152	351		
san200_0.7_1	200	13930	82		138	124	
san200_0.7_2	200	13930	60		134	115	
san200_0.9_1	200	17910	105		173	111	
san200_0.9_2	200	17910	105	106	175	119	
san200_0.9_3	200	17910	103	104	176	119	
san400_0.7_1	400	55860	113		277	253	
san400_0.7_2	400	55860	108		277	251	
san400_0.7_3	400	55860	80	82	274	248	
san400_0.9_1	400	71820	203		353	259	
sanr200_0.7	200	13868	67	68	137	125	
sanr200_0.9	200	17863	103	104	175	119	
sanr400_0.5	400	39984	74	75	201		
sanr400_0.7	400	55869	105	106	276	251	

The upper bounding techniques rely on partial linear relaxations of a b-coloring model, and are able to improve the best known bound for 49 of the 87 instances considered. The procedure devised to derive heuristic solution is based on the solution of four variations of a decisional model that takes in input an instance and a target value for the number of colors, and return either a feasible b-coloring solution with T colors, or a message stating that no feasible solution with such a number of colors exists. With a proper selection of the target number of nodes, the proposed methods were able to improve 17 heuristic solutions over the 87 instances considered. As a side effect, it was also possible to prove optimality for 11 instances for the first time.

Table 10
Improved bounds for the coloring instances from [12].

Instance			Lower Bounds		Upper Bounds		Newly Closed
Name	$\lvert V\rvert$	$\lvert E\rvert$	Old	New	Old	New	
dsjc125.9	125	3891	68		109	72	
dsjc250.9	250	27897	127	128	219	150	
DSJR500.1c	500	121275	142	150	478	221	
fpsol2.i.1	451	8691	77		79	77	*
inithx.i.1	864	18707	72		74	72	*
inithx.i.2	645	13979	50		52	50	*
inithx.i.3	621	13969	50		52	50	*
le450_5c	450	9803	35	36	52		
le450_15c	450	16680	54	55	93		
mulsol.i.1	197	3925	64		65	64	*
mulsol.i.2	188	3885	51		53	51	*
R125.1c	125	7501	53		116	53	*
R250.1c	250	30227	86		238	89	

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the anonymous reviewers for the careful reading and useful suggestions provided.

References

[1] Dominique Barth, Johanne Cohen, Taoufik Faik, On the b-continuity property of graphs, Discrete Appl. Math. 155 (13) (2007) 1761–1768.

[2] Victor A. Campos, Carlos V. Lima, Nicolas A. Martins, Leonardo Sampaio, Marcio C. Santos, Ana Silva, The b-chromatic index of graphs, Discrete Math. 338 (11) (2015) 2072–2079.

[3] Victor A. Campos, Carlos V. Lima, Ana Silva, B-coloring graphs with girth at least 8, in: Proceedings of the Seventh European Conference on Combinatorics, Graph Theory and Applications, 2013, pp. 327–332.

[4] Amal Dandashi, Mayez Al-Mouhamed, Graph coloring for class scheduling, in: ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010, 2010, pp. 1–4.

[5] Haytham Elghazel, Veronique Deslandres, Mohand-Said Hacid, Alain Dussauchoy, Hamamache Kheddouci, A new clustering approach for symbolic data and its validation: application to the healthcare data, Lect. Notes Comput. Sci. 4203 (2006) 373–482.

[6] Iztok Fister, Iztok Peterin, Marjan Mernik, Matej Črepinšek, Hybrid evolutionary algorithm for the b-chromatic number, J. Heuristics 21 (2015) 501–521.

[7] Djamel Gaceb, Veronique Eglin, Frank Lebourgeois, Hubert Emptoz, Improvement of postal mail sorting system, Int. J. Doc. Anal. Recognit. 11 (2008) 67–80.

[8] Djamel Gaceb, Veronique Eglin, Frank Lebourgeois, Hubert Emptoz, Robust approach of address block localization in business mail by graph coloring, Int. Arab J. Inf. Technol. 6 (3) (2009) 221–229.

[9] James S. Graham, Roberto Montemanni, Jim N.J. Moon, Derek H. Smith, Frequency assignment, multiple interference and binary constraints, Wirel. Netw. 14 (4) (2008) 449–464.

[10] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021.

[11] Robert W. Irving, David F. Manlove, The b-chromatic number of a graph, Discrete Appl. Math. 91 (1) (1999) 127–141.

[12] David S. Johnson, Michael A. Trick, Cliques, coloring, and satisfiability: second DIMACS implementation challenge, Am. Math. Soc. 26 (1996).
[13] Ivo Koch, Javier Marenco, An integer programming approach to b-coloring, Discrete Optim. 32 (2019) 43–62.
[14] Ivo Koch, Iztok Peterin, The b-chromatic index of direct product of graphs, Discrete Appl. Math. 190–191 (2015) 109–117.
[15] Jan Kratochvíl, Zsolt Tuza, Margit Voigt, The b-chromatic number of a graph, Lect. Notes Comput. Sci. 2573 (2002) 310–320.
[16] Enrico Malaguti, Paolo Toth, A survey on vertex coloring problems, Int. Trans. Oper. Res. 17 (1) (2010) 1–34.
[17] Rafael A. Melo, Michell F. Queiroz, Marcio C. Santos, A matheuristic approach for the b-coloring problem using integer programming and a multi-start multi-greedy randomized metaheuristic, Eur. J. Oper. Res. 295 (1) (2021) 66–81.
[18] Roberto Montemanni, Derek H. Smith, An iterative matheuristic algorithm for the B-coloring problem, in: The 3rd International Conference on Industrial Engineering and Industrial Management (IEIM 2022), 2022, pp. 265–270.
[19] Roberto Montemanni, Derek H. Smith, Stuart M. Allen, Lower bounds for fixed spectrum frequency assignment, Ann. Oper. Res. 107 (1) (2001) 237–250.
[20] Roberto Montemanni, Derek H. Smith, Stuart M. Allen, An improved algorithm to determine lower bounds for the fixed spectrum frequency assignment problem, Eur. J. Oper. Res. 156 (3) (2004) 736–751.
[21] Roberto Montemanni, Derek H. Smith, Andrea E. Rizzoli, Luca Maria Gambardella, Sequential ordering problems for crane scheduling in port terminals, Int. J. Simul. Process Model. 5 (4) (2009) 248–261.
[22] Derek H. Smith, Roberto Montemanni, Stephanie Perkins, The use of an exact algorithm within a tabu search maximum clique algorithm, Algorithms 13 (10) (2020).