A simple branching scheme for Vertex Coloring Problems

Stefano Gualandi[a], Federico Malucelli[b]

[a] *Università di Pavia, Dipartimento di Matematica, via Ferrata 1, Pavia*
`stefano.gualandi@unipv.it`
[b] *Politecnico di Milano, Dipartimento di Elettronica ed Informazione, Piazza L. da Vinci 32, Milano*
`malucell@elet.polimi.it`

Abstract

We present a branching scheme for some Vertex Coloring Problems based on a new graph operator called *extension*. The extension operator is used to generalize the branching scheme proposed by Zykov for the basic problem to a broad class of coloring problems, such as the graph multicoloring, where each vertex requires a multiplicity of colors, the graph bandwidth coloring, where the colors assigned to adjacent vertices must differ by at least a given distance, and graph bandwidth multicoloring, that generalizes both the multicoloring and the bandwidth coloring problems. We report some computational evidence of the new branching scheme effectiveness.

Key words: Graph Coloring, Graph Bandwidth-Multicoloring, Branching Scheme

1. Introduction

Given a graph $G = (V, E)$ and an integer k, a k-coloring of the graph G is a mapping of each vertex to a single color, such that adjacent vertices take different colors. The Minimum Graph Coloring Problem (Min–GCP), which is known to be NP-hard, consists in finding the minimum k such that a k-coloring exists. Such minimum k is the chromatic number of G and is denoted by $\chi(G)$, or simply by χ. A second graph coloring problem that generalizes Min–GCP asks to assign to every vertex a given number of colors, while satisfying the constraint that the same color cannot appear in adjacent vertices. This problem is known in the literature as the Minimum Graph Multicoloring Problem (Min–GMP), and though it can be reduced to Min–GCP on an auxiliary (much bigger) graph, it has its own interest and ad hoc algorithms can take advantage of its structure. Let $\chi_m(G)$ denote the minimum k for Min–GMP. A third version of the vertex coloring problem, defined on an edge-weighted graph, consists of assigning to each vertex a set of colors such that the distance between any color assigned to every pair of adjacent vertices is at least equal to the edge weight. This version of the problem is known in literature as the Bandwidth-Multicoloring Problem, or Bandwidth Coloring if every vertex requires a single color. Let $\chi_{mb}(G)$ and $\chi_b(G)$ denote the minimum number of colors k for Min–GBMP and for the simple Bandwidth Coloring problem, respectively. For a recent survey on Vertex Graph Coloring problems, see [1].

Branching schemes in implicit enumeration approaches as Branch&Bound, Branch&Cut and Branch&Price, are of utmost importance for the efficiency of the algorithms, especially when strongly structured problems are at hand. The addition of general branching constraints often risks to spoil the structure of the problem. Zykov in 1949 proposed a simple bipartite branching scheme for Min–GCP whose main feature is to generate in both branches two smaller instances of the same problem. For the other coloring problems, as Min–GMP, Min–GBMP and Bandwidth Coloring problem, to the best of our knowledge, no branching schemes with this characteristic have been proposed.

In the Branch&Price algorithm proposed in [2] for Min–GMP, the adopted branching schemes are quite involved and mainly fix variables corresponding to columns of the extended formulation to 0 or to 1. In the case of Frequency Assignment Problems (e.g., [3]), which basically are Min–GBMP problems on graphs with a special structure, the branching is carried out on binary variables that indicate the assignment of colors (frequencies) to vertices (cells). More involved branching schemes are also devised [4, 5], but they are always derived from the mathematical programming formulation of the problem.

In the following section, we introduce a new operator generalizing the basic Zykov branching scheme that exploits the problem structure, and maintains the graph size small. Some computational evidence of the branching scheme effectiveness is reported.

2. The extension operator and the branching scheme

Given a graph G and two non-adjacent vertices v and w, consider the two following operations: *contraction* and *insertion*. The contraction (or shrinking) operation, denoted by $G \setminus \{v, w\}$, consists of creating a new vertex u with an incident edge for each vertex in the union of the neighborhoods of v and w and and of removing vertices v and w from the vertex set. The insertion operation consists of adding the edge $\{v, w\}$ to E, and is denoted by $G + \{v, w\}$. As a corollary to a theorem in [6], we have:

Theorem 1. $\chi(G) = \min\{\chi(G \setminus \{v, w\}), \chi(G + \{v, w\})\}$ *[6].*

This well-known result has been exploited in a number of coloring algorithms that are usually called Zykov's (contraction) algorithms (e.g., [7, 8]). The branching scheme derived from this result considers two non adjacent vertices of G, v and w. The alternatives are: either v and w are assigned to different colors, enforced by the introduction of edge $\{v, w\}$ ($G + \{v, w\}$), or they are assigned to the same color, as a result of contraction of the two vertices $\{v, w\}$ ($G \setminus \{v, w\}$). In this case, note that the color of vertex u, corresponding to the color assigned to both v and w, must be different from the colors of vertices previously adjacent to v or to w. One of the best exact approach to graph coloring is to implement a branch-and-price algorithm, as pioneered in [9] and recently revived e.g. in [10], that implements a branching rule equivalent to the one derived from Theorem 1.

Let us consider now Min–GMP and Min–GBMP that are formally stated as follows.

Min–GMP. Given a graph $G = (V, E)$ and integer weights $c_v > 0$ on the vertices $v \in V$, assign c_v colors to each vertex v, so that the colors assigned to every pair of adjacent vertices v and w (i.e. $\{v, w\} \in E$) are all different. The objective is to minimize the total number of used colors.

Min–GBMP. Given a graph $G = (V, E)$ with integer weights $c_v > 0$ on the vertices $v \in V$ and integer weights $b_{vw} > 0$ on the edges, assign c_v colors to each vertex v, so that the colors assigned to every pair of adjacent vertices v and w (i.e. $\{v, w\} \in E$) are all different; moreover, identifying colors with natural numbers, the difference between any color assigned to v and any color assigned to w must be at least b_{vw}, for every $\{v, w\} \in E$. The objective is to minimize the bandwidth of the used colors, that is the difference between the highest and the smallest used color. In other cases the objective can be the minimization of the number of used colors.

We introduce now an operator that generalizes Zykov contraction and insertion described above. Consider two non adjacent vertices v and w of G, and let $N(v)$ be the neighborhood of v. Let $\uplus_t$ denote the new operator called *extension*, where t is an integer parameter having value between 0 and $\min\{c_v, c_w\}$.

Definition 1. *$G' = (V', E') = G \uplus_t \{v, w\}$ is obtained as follows:*

- $V' = V \cup \{u\}$, $c_u = t$.
- $c_v \leftarrow c_v - t$ *and* $c_w \leftarrow c_w - t$*;*
- E' *includes the following new edges, in addition to those of* E*:*
  - $\{v, w\}$, $\{v, u\}$, *and* $\{w, u\}$ *with weights* $b_{vw} = b_{vu} = b_{wu} = 1$*;*
  - $\{u, u'\}$ *with weight* $b_{uu'} = b_{vu'}$*, for each vertex* $u' \in N(v) \setminus N(w)$
  - $\{u, u'\}$ *with weight* $b_{uu'} = b_{wu'}$*, for each vertex* $u' \in N(w) \setminus N(v)$
  - $\{u, u'\}$ *with weight* $b_{uu'} = \max\{b_{vu'}, b_{wu'}\}$*, for each vertex* $u' \in N(v) \cap N(w)$*;*
- *finally all vertices* s *with* $c_s = 0$ *and all their incident edges are removed from* G'*.*

An application of the extension operator is exemplified in Figure 1. Note that the extension operator generalizes Zykov insertion and contraction. Consider the classical vertex coloring problem, equivalent to Min–GBMP where all weights c_v and b_{vw} are equal to one. In this case t is either 0 or 1: when $t = 0$, the operation $G \uplus_0 \{v, w\}$ is equivalent to the insertion of edge $\{v, w\}$, since the new vertex u has weight $c_u = 0$ and is omitted. When $t = 1$, the operation $G \uplus_1 \{v, w\}$ is equivalent to the contraction of v and w, since $c_v = c_w = 0$ and the vertices are removed.

By analogy with Zykov result, we can prove that the extension operator can be used within a branching scheme to decrease the complexity of the graph coloring instance, and eventually reduce to easy subproblems.

Theorem 2. *Consider an instance of* Min–GBMP *defined on graph G, two non adjacent vertices v and w, and let* $h = \min\{c_v, c_w\}$ *be the maximum number of colors that v and w can share, then*

$$\chi_{mb}(G) = \min_{t=0...h} \chi_{mb}(G \uplus_t \{v, w\}).$$

Proof. First of all notice that the set of feasible solutions of Min–GBMP is actually partitioned by considering the feasible solutions of the instances defined on $G \uplus_t \{v, w\}, t = 0, \dots, h$. Indeed either v and w have no colors in common, or they share one color, and so forth up to h colors. Now we show that for each of these cases the coloring feasibility is preserved thus implying the result.

$G \uplus_0 \{v, w\}$ has edge $\{v, w\}$ in addition with respect to G and $b_{vw} = 1$, thus obliging the problem instance to have no shared colors between v and w, without any further constraint on the bandwidth. Thus, the feasible region of Min–GBMP for $G \uplus_0 \{v, w\}$ contains all feasible solutions on G with no shared colors between v and w.

In $G \uplus_h \{v, w\}$ at least one of the two vertices v and w is removed since c_v or c_w is set to 0. Moreover vertex u is added with a request of $c_u = h$ colors, that is those that must be shared between v and w in G. To preserve the coloring feasibility, u is made adjacent to all the neighbors of v and w in G; the bandwidth requirement is significant only for those vertices u' adjacent to both v and w, and in this case the bandwidth requirement is set to the most restrictive between $b_{vu'}$ and $b_{wu'}$. Also in this case, the feasible region of Min–GBMP for $G \uplus_h \{v, w\}$ contains all feasible solutions on G with exactly h shared colors between v and w.

Consider now $G \uplus_t \{v, w\}$, with $0 < t < h$. In this case neither v nor w is deleted from G, but their number of required colors is decreased by t units, corresponding to the number of colors that must be different from those of the other vertex (requirement enforced by the presence of the new edge $\{v, w\}$). As in the previous case we introduce the new vertex u with a request of $c_u = t$ colors, corresponding to those shared by v and w. To preserve the feasibility in the assignment of the shared colors, also in this case we introduce the edges between the neighbors of v and w with suitable bandwidth requirements. It is easy to note that the feasible region of Min–GBMP for $G \uplus_t \{v, w\}$ contains all feasible solutions of Min–GBMP on G with exactly t shared colors between v and w. □

Now we shall prove that by applying the extension operator we actually generate simpler subproblems and that by repeatedly applying it we eventually converge to trivial instances.

Theorem 3. *Consider the branching scheme that from an instance of* Min–GBMP *defined on graph G and two non adjacent vertices v and w, with* $h = min\{c_v, c_w\}$ *generates the instances* $G \uplus_0 \{v, w\}, G \uplus_1 \{v, w\}, \dots, G \uplus_h \{v, w\}$. *The repeated application of the branching scheme converges toward instances of* Min–GBMP *defined on cliques.*

Proof. Consider the following instances generated by the branching:

(a) $G \uplus_0 \{v, w\}$: in this case the operator is equivalent to the graph insertion $G + \{v, w\}$, so the new graph has one additional edge.

(b) $G \uplus_t \{v, w\}, 0 < t < h$ (when $h > 1$): in this case both vertices v and w have their weights decreased by t units, and the new vertex u has a degree equal to $|N(v) \cup N(w)| + 2$. Hence the weights of two vertices of G decrease of t units and a new node with weight t is created, with an overall balance of $-t$ in the sum of the nodes weights.

(c) $G \uplus_h \{v, w\}$: in this case at least one vertex, v or w, is removed and the weight of the possibly remaining one decreases by h units (if $c_v = c_w$ both v and w are removed). The new vertex u has a degree equal to $|N(v) \cup N(w)|$, in case both v and w are removed, or equal to $|N(v) \cup N(w)| + 1$ in case only one is removed, but it has a weight equal to h. Therefore, overall balance of the node weights is negative, the number of nodes either decreases and the number of edges may not increase, or the number of nodes remains unchanged and the number of edges strictly increases.

Note that in all instances generated by the branching the graph is denser (cases (a) and (c)) with respect to the original one, or the weights of the nodes decrease. This means that, by recursively applying the branching, we eventually obtain Min–GBMP instances defined on cliques, or instances with unit weights, corresponding to single coloring problems. For these instances the proposed branching scheme corresponds to the classical Zykov one, thus the convergence to clique instances follows immediately. □

We now show how to color vertices in a Min–GBMP instance with weights on nodes and arcs where the graph G is a clique. Note that Min–GMP instances defined on cliques need a number of different colors equal to the sum of the vertex weights c_v. Since the graph is a clique it is sufficient to assign colors to vertices according to the required number. In the case of Min–GBMP, besides the multiplicity of colors required in the vertices, we have to care about the bandwidth requirements on edges. We can proceed as follows:

- Sort the vertices in any order (without loss of generality, we may use the vertex numbering from 1 to n) and assign a direction to edges so that if $v < w$ the direction of edge $\{v, w\}$ is (v, w). Let A denote the set of directed arcs.
- Solve the following problem:

$$\begin{aligned} \min \quad & \pi_n - \pi_1 \\ \text{s.t.} \quad & \pi_v + c_v + b_{vw} \geq \pi_w \quad \forall (v, w) \in A \end{aligned}$$

- To each vertex v assign c_v consecutive colors starting from color π_v.

Note that the above linear programming problem is a potential problem and can be solved by looking for the longest path in the directed graph $\bar{G} = (V, A)$ where the arc lengths are given by $c_v + b_{vw}$ for every arc (v, w) of A. Since the graph is acyclic, due to the direction assigned to the arcs that satisfies the topological order, this computation can be done in linear time.

3. Computational Results

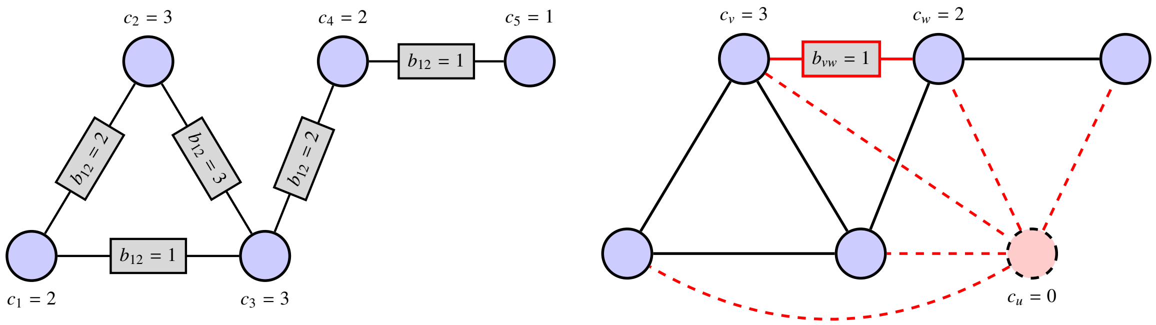
In order to assess the strength of the proposed branching scheme, we have implemented a branch-and-price algorithm for solving the Graph Multicoloring Problem, and we have experimented with a set of benchmark instances taken from those of the DIMACS challenge. The results of our branching scheme are compared with those obtained by applying the same algorithm to the equivalent Min–GCP instances obtained by expanding the graph. Each vertex v with $c_v > 1$ is substituted by a clique of c_v nodes each connected to the neighbors of v. In table 1, n and m are the number of vertices and edges of the original graph and n' and m' are the number of vertices and edges in the equivalent expanded graph. UB_0 is the upper bound found at the root node. We report the results only for six instances because for all the other ones the branch-and-price applied to the original graph finds the optimal solution at the root node. The results of the other instances can be found in [10]. We limit the node number of the enumeration tree ("Nodes" in the table) to 500. The branch-and-price applied to the expanded graph never reaches optimality in 500 nodes. We report the best lower and upper bounds (LB and UB, respectively) found by the algorithms. The difference is 0 when the optimality is proved. While the lower bound values are equal, the upper bounds provided by the branch-and-price using the new branching scheme are in most of the instances smaller. Moreover the overall execution time T to reach the termination (either due to optimality or to node limit) and the average execution time per explored node T/N are impressively better.

References

[1] E. Malaguti, P. Toth, A survey on vertex coloring problems, Internat. Trans. Oper. Res. 17 (1) (2010) 1–34.

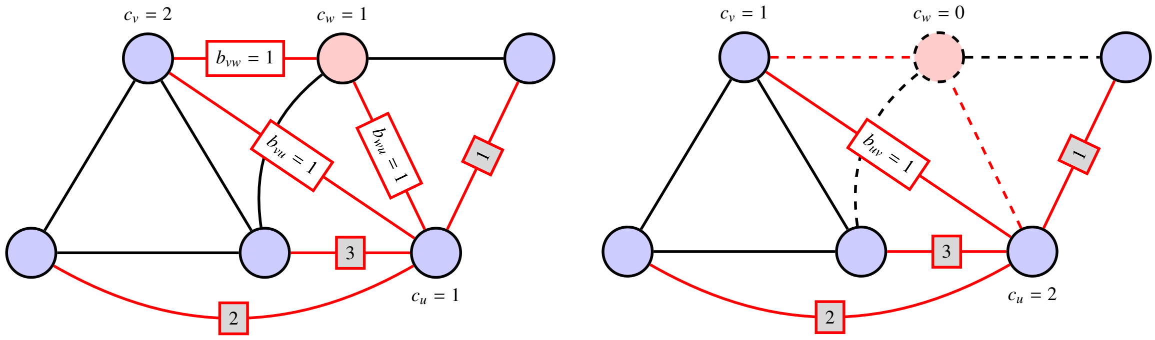
[2] A. Mehrotra, M. Trick, A branch-and-price approach for graph multi-coloring, Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies. (2007) 15–29.

[3] K. I. Aardal, C. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, A. Sassano, Models and solution techniques for the frequency assignment problem, 4OR: Quart. J. Oper. Res. 1 (4) (2003) 261–317.



(a) Initial graph G: select vertices $v = 2$ and $w = 4$ having $c_v = 3$ and $c_w = 2$. Therefore, $h = 2$ is the maximum number of shared colors between v and w. The labels on the edges indicate the required bandwidth.

(b) $G \uplus_0 \{v, w\}$: The vertices v and w do not share any color, and the edge $\{v, w\}$ gets weight $b_{vw} = 1$. Vertex u along with its incident edges is omitted, since it has $c_u = 0$.



(c) $G \uplus_1 \{v, w\}$: The vertices v and w share one color: we set $c_v = 2$, $c_w = 1$, and $c_u = 1$. Each edge incident in u gets a weight as specified in Definition 1.

(d) $G \uplus_2 \{v, w\}$: The vertices v and w share two colors: we set $c_v = 1$, $c_w = 0$, and $c_u = 2$. Vertex w, having $c_w = 0$, is omitted.

Figure 1: Example of applying the $\uplus_t$ operator to an instance of Min–GBMP: (a) the initial graph, and (b)–(d) the three branching nodes, when the vertices 2 and 4 are selected, corresponding to $G \uplus_0 \{v, w\}$, $G \uplus_1 \{v, w\}$, and $G \uplus_2 \{v, w\}$, respectively.

Table 1: Branching rules: comparison with a limit of visiting 500 hundred vertices.

		New Branching Rule							Branching on the Expanded Graph						
Instance	UB_0	n	m	LB	UB	Nodes	Time	T/N	n'	m'	LB	UB	Nodes	Time	T/N
queen8_8g	29	64	728	28	28	7	2	0.3	185	6195	28	29	500	2833	5.7
R75_1g	15	70	251	14	14	8	1	0.1	216	2757	14	15	500	4384	8.8
R100_5g	43	100	2456	42	43	500	98	0.2	296	21905	42	43	500	2690	5.4
queen10_10g	39	100	1470	38	38	135	124	0.9	293	12980	38	39	500	3176	6.4
queen11_11gb	144	121	1980	140	140	306	301	1.0	1258	219078	140	144	500	42494	85.0
queen11_11g	45	121	1980	41	42	500	756	1.5	362	17977	41	45	500	3528	7.1

[4] M. Fischetti, C. Lepschy, G. Minerva, G. Romanin-Jacur, E. Toto, Frequency assignment in mobile radio systems using branch-and-cut techniques, Eur. J. Oper. Res. 123 (2) (2000) 241–255.

[5] T. Hemazro, B. Jaumard, O. Marcotte, A column generation and branch-and-cut algorithm for the channel assignment problem, Comp. & OR 35 (4) (2008) 1204–1226.

[6] A. A. Zykov, On some properties of linear complexes, Mat. Sb. 24(66) (2) (1949) 163–188.

[7] D. Corneil, B. Graham, An algorithm for determining the chromatic number of a graph, SIAM J. Comput. 2 (1973) 311.

[8] C. McDiarmid, Determining the chromatic number of a graph, SIAM J. Comput. 8 (1) (1979) 1–14.

[9] A. Mehrotra, M. Trick, A column generation approach for graph coloring, INFORMS J. Comput. 8 (4) (1996) 344–354.

[10] S. Gualandi, F. Malucelli, Exact solution of graph coloring problems via constraint programming and column generation, published online in Articles in Advance, February 2, 2011 (2011). doi:10.1287/ijoc.1100.0436.