A Branch-and-Cut Algorithm for the Partition Coloring Problem

Yuri Frota
COPPE/UFRJ, Programa de Engenharia de Sistemas e Otimização
Rio de Janeiro, RJ 21945-970, Brazil
abitbol@cos.ufrj.br

Nelson Maculan
COPPE/UFRJ, Programa de Engenharia de Sistemas e Otimização
Rio de Janeiro, RJ 21945-970, Brazil
maculan@cos.ufrj.br

Thiago F. Noronha
Catholic University of Rio de Rio de Janeiro, Department of Computer Science
Rio de Janeiro 22453-900, Brazil
tfn@inf.puc-rio.br

Celso C. Ribeiro
Universidade Federal Fluminense, Department of Computer Science
Niterói, RJ 22410-240, Brazil
celso@inf.puc-rio.br

Keywords: Partition coloring problem, branch-and-cut, optical networks

1. Partition coloring

Let $G = (V, E)$ be an undirected graph, where E is the set of edges and V is the set of vertices. Furthermore, let $Q = \{Q_1, \ldots, Q_q\}$ be a partition of V into q subsets such that $Q_1 \cup \ldots \cup Q_q = V$ and $Q_i \cap Q_j = \emptyset$, for all $i, j = 1, \ldots, q$ with $i \neq j$. We refer to $Q_1, \ldots, Q_q$ as the *components* of the partition, while $P[v]$ denotes the index of the component of vertex $v \in V$. The *Partition Coloring Problem* (PCP) consists in finding a subset of vertices $V' \subseteq V$ such that $|V' \cap Q_i| = 1, i = 1, \ldots, q$ (i.e., V' contains one vertex from each component $Q_i, i = 1, \ldots, q$), and the chromatic number of the graph induced in G by V' is minimum. This problem is clearly a generalization of the graph coloring problem. Li and Simha [6] have shown that the decision version of PCP is NP-complete.

Algorithms for solving PCP have been used in the literature as building blocks for algorithms for the Routing and Wavelength Assignment (RWA) problem in optical networks. In such networks, each signal is converted to the optical domain and reaches the receptor without conversion to the electrical domain. Wavelength Division Multiplexing (WDM) allows the more efficient use of the huge capacity of optical fibers, as far as it permits the simultaneous transmission of different channels along the same fiber, each of them using a different wavelength. An all-optical point-to-point connection between two vertices is called a lightpath. Two lightpaths may use the same wavelength, provided they do not share any common link. The routing and wavelength assignment problem consists in routing a set of lightpaths and assigning a wavelength to each of them. Variants of RWA are characterized by different optimization criteria and traffic patterns, see e.g. [2].

Li and Simha [6] proposed a two phase decomposition strategy to solve RWA, in which the objective function consisted in minimizing the total number of wavelengths used to route all traffic demands.

In the first phase, one or more possible routes are computed for each lightpath. In the second, one pre-computed route and one wavelength are assigned to each lightpath by a greedy heuristic applied to an associated PCP instance. In this transformed instance, the vertices correspond to routes, there is an edge between each pair of vertices whose associated routes share a common link, and all alternative routes associated to the same connection are placed in the same component of the partition. Noronha and Ribeiro [8] followed the same decomposition strategy, but used different algorithms in each phase. The set of alternative routes was pre-computed by using the k−EDR procedure [8], while the PCP instance was solved by a tabu search heuristic. Jaumard et al. [5] considered the objective of minimizing the blocking rate and extended the strategy proposed in [8].

This paper presents an exact algorithm for the partition coloring problem. A generalization of the 0-1 formulation for the vertex coloring problem proposed in [1, 3] is described. The linear relaxation of the proposed formulation is used to provide lower bounds. A tabu search heuristic provides feasible solutions and upper bounds. We also propose a cutting plane procedure to strength the lower bounds and a decomposition strategy for solving PCP. Section 2 summarizes the proposed branch-and-cut algorithm. Computational experiments are reported in Section 3. Concluding remarks are drawn in the last section.

2. Branch-and-cut

In this section, we describe the branch-and-cut algorithm for PCP. We first present an integer programming formulation based on a generalization of the 0-1 formulation proposed in [1, 3] for the graph coloring problem. Next, we describe the branching strategy to decompose PCP in two subproblems. Following, we briefly describe the valid inequalities that are used in a cutting plane procedure developed for improving the linear relaxation bound. Finally, we show how to build good feasible solutions for each subproblem in the branch-and-cut tree.

A coloring of a graph $G = (V, E)$ can be viewed as a family $W_1, \ldots, W_k$ of independent sets, with $k \geq \chi(G)$, where $\chi(G)$ is the chromatic number of the graph. Each independent set in this family is associated with a different color. For each color $i = 1, \ldots, k$, we choose a vertex to represent the corresponding independent set W_i. Then, each colored vertex can be in one of the following two states: either it represents the vertices colored with its color or there exists another colored vertex that represents its color. We define $A(u) = \{w \in V : (u, w) \notin E, w \neq u\}$ as the anti-neighborhood of vertex u (i.e., the subset of vertices that are not adjacent to u) and $A_P(u) = \{v \in A(u) : P[u] \neq P[v]\}$ as the *component anti-neighborhood* of a vertex $u \in V$ (i.e., the vertices in the anti-neighborhood of u that are in another component of the partition). We also define $A'_P(u) = A_P(u) \cup \{u\}$. Given a subset of vertices $V' \subseteq V$, we denote by $E[V']$ the subset of edges induced in $G = (V, E)$ by V'. A vertex $v \in A_P(u)$ is said to be *isolated* in $A_P(u)$ if $E[A_P(u)] = E[A_P(u) \setminus \{v\}]$ (i.e., vertex v has no adjacent vertex in $A_P(u)$). We define the binary variables x_{uv} for all $u \in V$ and for all $v \in A'_P(u)$, such that $x_{uv} = 1$ if and only if vertex u represents the color of vertex v; $x_{uv} = 0$ otherwise. The PCP can be formulated as the following integer programming problem:

$$\min \sum_{u \in V} x_{uu} \tag{1}$$

subject to:

$$\sum_{u\in Q_p}\sum_{v\in A'_P(u)} x_{vu} \geq 1 \quad \forall p = 1,\ldots,q \tag{2}$$

$$x_{uv} + x_{uw} \leq x_{uu} \quad \forall u \in V, \quad \forall (v,w) \in E \text{ with } v,w \in A_P(u) \text{ and } P[v] \neq P[w] \tag{3}$$

$$x_{uv} \leq x_{uu} \quad \forall u \in V, \quad \forall v \in A_P(u) \text{ such that } v \text{ is isolated in } A_P(u) \tag{4}$$

$$x_{uv} \in \{0,1\} \quad \forall u \in V, \quad \forall v \in A'_P(u). \tag{5}$$

The above model is said to be the *formulation by representatives* of PCP. The objective function (1) counts the number of representative vertices, i.e., the number of colors. Constraints (2) enforce that each component $Q_p, p = 1,\ldots,q$, must have at least one of its vertices $u \in Q_p$ represented either by itself ($x_{uu} = 1$) or by some other vertex v ($x_{vu} = 1$, with $v \neq u$) in its component anti-neighborhood (i.e., a vertex from another component that do not share an edge with u). Inequalities (3) enforce that adjacent vertices have distinct representatives. Inequalities (3) together with constraints (4) ensure that a vertex can only be represented by a representative vertex.

To break symmetries in the above formulation, we also generalized the asymmetric formulation by representatives in [1]. An order $\ll$ is defined between the components. A vertex u can only represent a vertex v if $Q_{P[u]} \ll Q_{P[v]}$. In this case, $x_{uv} = 1$; otherwise $x_{uv} = 0$. For sake of conciseness, we ommit here the asymmetric formulation.

The classical Mehrotra and Trick's branching rule [7] for the graph coloring problem branches on two non-adjacent vertices. We adopted a modified rule which takes into account the specific characteristics of PCP. We define the following operations for PCP, for any $i,j = 1,\ldots,p$ such that $i \neq j$: SAME(i,j) enforces that the same color is used to color components Q_i and Q_j, while DIFFER(i,j) requires that different colors are used to color components Q_i and Q_j. The first operation can be implemented by merging the two components into a single one: the new component will be formed by merging every pair of vertices w and z into a new vertex x, with $w \in Q_i$, $z \in A_P(w) \cap Q_j$, and the neighbors of x being those of w or z. DIFFER(i,j) can be implemented by inserting edges between every pair of non-adjacent vertices in components Q_i and Q_j. These operations create two new PCP subproblems, such that an optimal feasible coloring exists in exactly one of them. In addition, this branching rule has the advantage of leading to denser graphs without increasing the complexity of the subproblems.

We define $\overline{G} = (\overline{V}, \overline{E})$ as the graph associated to the current subproblem of the branch-and-cut tree. To improve the linear relaxation bound, we generalized the original family of valid inequalities [1, 3] based on two different kinds of cuts. *External cuts* are based on the idea of how many of the vertices of a subset $K \subseteq A_P(u)$ with a particular structure (such as cliques, holes, and anti-holes) can be represented by vertex u, This idea leads to the inequality $\sum_{v\in K} x_{uv} \leq \alpha_K x_{uu}$, where α_K is the size of a maximum independent set of K. *Internal cuts* are based on the idea of how many colors will be necessary to color any odd hole or anti-hole of the graph. Except for *elementary vertices* that stand alone in their components, no vertex has any guarantee of being colored in a feasible solution to PCP. Therefore, $\sum_{v\in H}(x_{vv} + \sum_{u\in A_P(v)\setminus H} x_{uv}) \geq \chi(H)$, where $H \subseteq \overline{V}$ is an odd hole or anti-hole compound of elementary vertices and $\chi(H)$ is the chromatic number of the subgraph induced by H in $\overline{G}$. The separation problem for PCP consists basically in finding cliques, holes, and anti-holes in $\overline{G}$. We used a GRASP heuristic for finding clique cuts and a modification of Hoffman's heuristic [4] for finding odd holes and anti-holes cuts.

The tabu search procedure TS−PCP proposed in [8] provides an upper bound to each node of the branch-and-cut tree. However, we propose a different procedure to provide an initial feasible solution to TS−PCP. Since the branching strategy described in the previous paragraph implies that the graphs associated to parent vertices differ from those associated to their children by only two components, we can construct an initial feasible solution to TS−PCP at any node of the branch-and-cut tree by using the solution associated to its parent node. The constructive procedure starts from a partial solution that is equal to the solution of the parent node in the branch-and-cut tree, except for the two components involved in the branching operation. The partial solution has at most two uncolored components that are later colored by algorithm onestepCD [6]. Therefore, TS−PCP starts from an initial solution with at most two more colors than the solution at the parent node.

3. Computational results

Algorithm B&C−PCP described in the previous section was implemented in C++ and compiled with version v3.41 of the Linux/GNU compiler. The linear relaxation of the integer programming formulation was solved by XPRESS version 2005-a. All experiments were performed on an AMD-Atlon machine with a 1.8 GHz clock and one Gbyte of RAM memory.

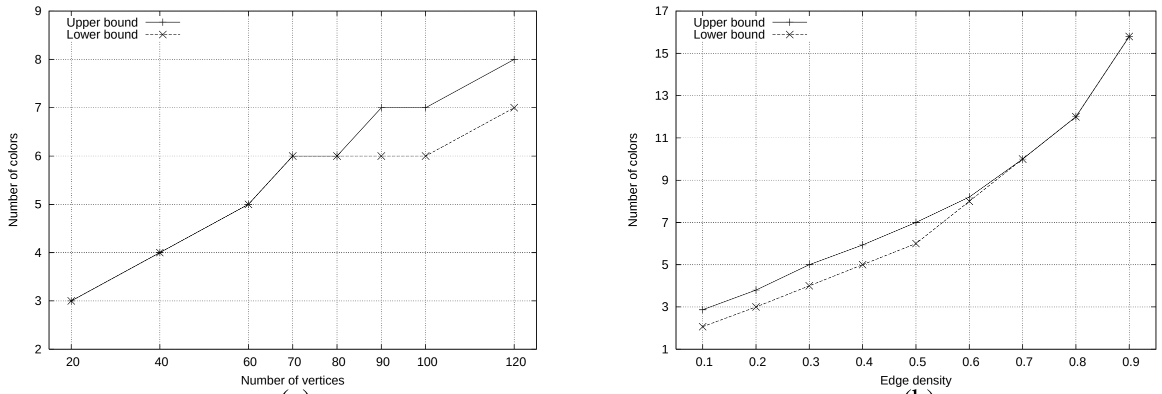
We first investigated the behavior of B&C−PCP for randomly generated graphs with 20 to 120 vertices. Each component has exactly two vertices and the edge density is 0.5. Five graph instances were generated for each number of vertices. Algorithm B&C−PCP was run three times for each instance, with different seeds for the pseudo-random number generator. The algorithm stopped when the optimal solution was found or after two hours of processing time. The average lower and upper bounds for the number of colors over the 15 runs for each graph size are plotted in Figure **??**. Algorithm B&C−PCP solved to optimality all instances with up to 80 vertices. Regarding the larger instances, the gap between the lower and upper bounds was always exactly equal to one color.

In the second experiment, we randomly generated graphs with 90 vertices distributed over 45 components with two vertices each. The edge density ranged from 0.1 to 0.9. Five graph instances were generated for each density. Algorithm B&C−PCP was run three times with different seeds for each instance. As for the first experiment, the algorithm stopped when the optimal solution was found or after two hours of processing time. The average lower and upper bounds for the number of colors over the 15 runs for each graph size are plotted in Figure **??**. The number of instances solved to optimality for each edge density is plotted in Table 1. The most difficult instances are those with edge density from 0.3 to 0.5. The number of instances solved to optimality increases with the edge density, because graphs with higher edge densities have more larger clicks that lead to better lower bounds. We also observe that the gap between the lower and upper bounds was never larger than one color in the instances not solved to optimality.

Edge density	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Instances solved to optimality	5	3	—	1	—	12	15	15	15

Table 1: Instances solved to optimality (out of 15) for different edge densities.

Finally, we report computational results for PCP instances arising from those of the routing and wavelength assignment problem. We consider the topology of the real network NSFnet with 14 vertices and 21 links, widely used for computational experiments in the literature. All links are



(a) (b)

Figure 1: Average lower and upper bounds for the number of colors: (a) varying the number of vertex; (b) varying the edge density.

bidirectional and the lightpaths are not symmetric [8], i.e., the lightpath from vertex i to j may be different from that from vertex j to i. We first generated RWA instances in which a lightpath connecting vertex i to j should be established with probabilities ranging from 0.1 to 1.0 by steps of 0.1. Five RWA instances were generated for each probability value. Every RWA instance was transformed into a PCP instance by the `2-EDR` procedure proposed in [8]. First, `2-EDR` computes up to two alternative routes for each lightpath. Then, a PCP instance is build with one vertex for each alternative route. There is one edge between each pair of vertices whose associated routes share a common link. All vertices associated to the same lightpath are placed in the same component of the partition.

Algorithm `B&C-PCP` was applied to each of the 50 above instances. The algorithm was made to stop whenever the optimal solution was found or after two hours of processing time. The computational results are presented in Table 2. The first column displays the probability of the existence of each lightpath. The three next columns give the average number of vertices, edges, and components of the five PCP instances. The next columns display the average and the maximum number of evaluated nodes in the `B&C-PCP` tree, the average and the maximum absolute gaps $\lceil UB - LB \rceil$ (where UB and LB denote, respectively, the best upper and lower bounds for each instance), the average and the maximum relative gaps $(UB - LB)/LB$, and the number of instances solved to optimality (over the five instances associated with the same ligthpath probability). All instances in which the probability of the existence of a lightpath request is less than or equal to 0.5 were solved to optimality at the root node of the branch-and-cut tree. Furthermore, the lower bound provided by rounding up the objective value of the linear programming solution of each of these instances at the root node of the `B&C-PCP` is already the optimal value. Although denser instances are considerably harder, approximately half of them could be solved and the average gaps were never larger than 5%.

4. Concluding remarks

In this work, we proposed (i) an integer programming formulation for the partition coloring problem based on the model of representatives, (ii) valid inequalities and cutting plane heuristics, (iii) a primal constructive heuristic, (iv) a branching strategy, and (v) a branch-and-cut algorithm for solving

Prob.	Av. vertices	Av. edges	Av. comps.	Av. (max.) nodes	Av. (max.) gap	Av. (max.) % gap	Solved
0.1	26.4	36.0	22.2	1 (1)	– (–)	– (–)	5
0.2	52.6	149.0	39.0	1 (1)	– (–)	– (–)	5
0.3	79.0	359.6	58.6	1 (1)	– (–)	– (–)	5
0.4	103.0	634.4	75.4	1 (1)	– (–)	– (–)	5
0.5	125.2	947.6	92.0	1 (1)	– (–)	– (–)	5
0.6	151.2	1410.4	109.6	24.2 (75)	0.4 (1)	4 (11)	3
0.7	181.0	2093.8	130.4	4.0 (16)	0.2 (1)	2 (10)	4
0.8	204.6	2638.2	146.6	8.6 (18)	0.6 (1)	5 (9)	2
0.9	224.8	3118.2	165.6	4.6 (8)	0.6 (1)	5 (8)	2
1.0	248.8	3885.6	182.0	2.2 (4)	0.4 (1)	3 (8)	3

Table 2: Computational results for RWA instances associated with NSFnet.

the partition coloring problem.

The computational experiments were carried out on randomly generated graph instances and on PCP instances arising from the problem of routing and wavelength assignment in all optical WDM networks. Instances with up to 251 nodes and 3973 edges were solved to optimality. Regarding the instances not solved to optimality within the 2-hour time limit, the absolute gaps were never larger than one color.

References

[1] M. Campêlo, V. Campos, and R. Corrêa. On the asymmetric representatives formulation for the vertex coloring problem. In *Proceedings of the 2th Brazilian Symposium on Graphs, Algorithms and Combinatorics*, volume 19 of *Electronic Notes in Discrete Mathematics*, pages 337–343, Angra dos Reis, 2005.

[2] J.S. Choi, N. Golmie, F. Lapeyrere, F. Mouveaux, and D. Su. A functional classification of routing and wavelength assignment schemes in DWDM networks: Static case. In *Proceedings of the 7th International Conference on Optical Communication and Networks*, pages 1109–1115, Paris, 2000.

[3] R. C. Corrêa, M. Campêlo, and Y. A. M. Frota. Cliques, holes and the vertex coloring polytope. *Information Processing Letters*, 89:159–164, 2004.

[4] K. L. Hoffman and M. Padberg. Solving airline crew scheduling problems. *Management Science*, 39:657–682, 1993.

[5] B. Jaumard, C. Meyer, and X. Yu. How much wavelength conversion allows a reduction in the blocking rate? Technical Report G-2006-05, GERAD, Montréal, 2006.

[6] G. Li and R. Simha. The partition coloring problem and its application to wavelength routing and assignment. In *Proceedings of the First Workshop on Optical Networks*, Dallas, 2000.

[7] A. Mehrotra and M. A. Trick. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8:344–354, 1996.

[8] T. F. Noronha and C. C. Ribeiro. Routing and wavelength assingn by partition coloring. *European Journal of Operational*, 171:797–810, 2006.