数理解析研究所講究録
第2069巻 2018年 84-94

A BOUNDING ALGORITHM FOR SELECTIVE GRAPH COLORING PROBLEM

YOICHI IZUNAGA AND KEISUKE SATO

ABSTRACT. This note addresses the selective graph coloring problem, which is a generalization of the well-known vertex coloring problem. Given an undirected graph together with a partition of its vertex set, it is to find a subset of the vertex set which shares exactly one vertex with each component of the partition so that the chromatic number of the subgraph induced by the subset is minimum. In this note, we present a new column generation algorithm for a linear programming relaxation problem of the selective graph coloring.

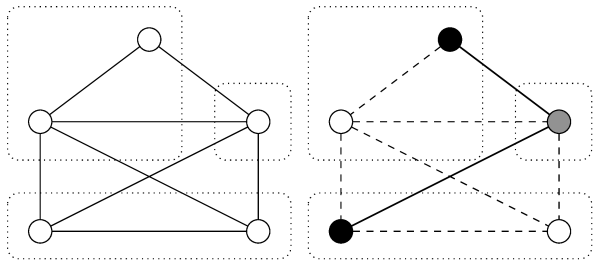
1. INTRODUCTION

Vertex coloring problem (VCP) is one of the most studied combinatorial optimization problem and has a wide range of applications. Given an undirected graph $G = (V, E)$ with the set of vertices $V = \{1, 2, \ldots, n\}$ and the set of edges E, the objective of VCP is to find an assignment of colors to each vertex of the graph such that no adjacent vertices receive the same color and the number of colors is minimized. The minimum number of colors is called *chromatic number* of G, denoted by $\chi(G)$.

Several variants of VCP have been proposed so far, e.g., vertex multi-coloring [11], weighted vertex coloring [6], and robust vertex coloring [12], in order to reflect more complex situations in real world applications. In this note, we focus on *selective graph coloring problem* (SCP), which is also called *partition coloring problem* [4]. In this problem, we are given an undirected graph as well as a k-partition $Q = \{V_1, V_2, \ldots, V_k\}$ of its vertex set V, and then the objective of SCP is to find a subset $V^\star \subseteq V$ which shares exactly one vertex with each component V_i of Q so that the chromatic number of the subgraph induced by $V^\star$ is minimum. Throughout the paper, we refer to the subset $V^\star$ and the chromatic number of the induced subgraph $G(V^\star)$ as *selective coloring* and *selective chromatic number* (denoted by $\chi(G, Q)$), respectively.

SCP originates in an application to the routing and wavelength assignment problem [9], and then spreads over various fields in real world, including dichotomy-based constraint encoding, antenna positioning and frequency assignment, timetabling problem, quality test scheduling, berth allocation problem, and multiple stacks traveling salesman problem. See Demange et al. [3] for details.

Obviously, SCP is an NP-hard optimization problem since it includes VCP as a special case where the partition Q consists of all singletons of V. For such a hard optimization problem, bounding procedures, in terms of both upper and lower bounds, play an important role. In particular, solution methods for the problems involving some integer variables rely heavily on bounding in the branch-and-bound framework. In order to obtain a lower bound



(A) a graph G and a partition Q (dotted-line)

(B) a selective coloring of G with two colors

FIGURE 1. Example.

on the selective chromatic number of SCP, we investigate a column generation algorithm based on the work by Coniglio and Tieves [2], which is regarded as a cutting plane algorithm in the dual space via a criterion of maximizing the bound improvement.

This note is organized as follows. In Section 2, we present the set covering formulation of the selective graph coloring problem and its linear programming relaxation problem. In Section 3, after reviewing the column generation algorithm, and we introduce the recently proposed bound-optimal generation. In Section 4, we propose an improved algorithm based on the bound-optimal generation. Lastly, in Section 5, we report computational experiments on the proposed algorithm.

2. Set Covering Formulation and its LP Relaxation

In the literature, various formulations for SCP have been proposed. Frota et al. [5] proposed a formulation for SCP based on the idea of the representative vertices presented by Campelo et al. [1], which avoids the symmetry. Hoshino et al. [8] presented a formulation which combines the asymmetric representative formulation and the set covering formulation by means of family of stable sets. More recently, Furini et al. [7] presented a simpler set-covering based formulation, and proposed a branch-and-price algorithm.

In this study, we adopt the set covering formulation provided by Furini et al. [7]. First, let us introduce the family of stable sets which shares at most one vertex with each component of the partition, i.e.,

$$\mathcal{S} = \{S \subseteq V \mid \{u, v\} \notin E \text{ for any } u, v \in S \text{ such that } |S \cap V_i| \leq 1 \text{ for } i \in K := \{1, 2, \ldots, k\}\}.$$

Introducing a binary variable z_S defined as

$$z_S = \begin{cases} 1 & \text{if vertices in } S \text{ is assigned the same color,} \\ 0 & \text{otherwise,} \end{cases}$$

for $S \in \mathcal{S}$, then SCP is formulated as the following set covering problem:

$$\text{P:} \quad \left| \begin{array}{lll} \text{minimize} & \sum_{S \in \mathcal{S}} z_S & \\ \text{subject to} & \sum_{S \in \mathcal{S}} a_{iS} z_S \geq 1 & \text{for } i \in K, \\ & z_S \in \{0,1\} & \text{for } S \in \mathcal{S}, \end{array} \right.$$

where $a_{iS} = 1$ when $|S \cap V_i| = 1$ and $a_{iS} = 0$ otherwise, for any $i \in K$ and $S \in \mathcal{S}$. We will call the first set of constraints *set covering constraint*. It should be noted that any solution of the above problem can be transformed into a selective coloring without changing its objective value. Throughout the paper, let us denote the optimal value of an optimization problem $(\cdot)$ by $\omega(\cdot)$.

To obtain a lower bound on the optimal value of SCP, we solve the *Linear Programming relaxation*, LP relaxation for short, where the binary constraint $z_S \in \{0,1\}$ is replaced by $0 \leq z_S \leq 1$. It is given as

$$\text{MP:} \quad \left| \begin{array}{lll} \text{minimize} & \sum_{S \in \mathcal{S}} z_S & \\ \text{subject to} & \sum_{S \in \mathcal{S}} a_{iS} z_S \geq 1 & \text{for } i \in K, \\ & z_S \geq 0 & \text{for } S \in \mathcal{S}. \end{array} \right.$$

The upper bound constraint on z_S is redundant, hence it can be omitted. This problem is referred to as *master problem* (MP).

3. Column Generation

3.1. **Standard column generation.** Owing to the presence of exponentially many variables in the formulations mentioned in Section 2, we can not feed them to general purpose optimization solvers, e.g., CPLEX, Xpress, and Gurobi as the size of instance is large. For the problems of this kind, it is natural to rely on column generation to solve them.

Column generation operates with a *restricted master problem* (RMP) which includes only a small number of variables of the whole. At each iteration in the column generation, we solve RMP to optimality at first, and then either try to generate a column (or variable) with a negative reduced cost or to prove that no such column exists. This latter process at each iteration results in solving the *column generation subproblem*.

We give a detailed description of all components in the column generation below. Given a subfamily $\mathcal{C} \subseteq \mathcal{S}$, we consider the following restricted master problem:

$$\text{RMP}(\mathcal{C}): \quad \left| \begin{array}{lll} \text{minimize} & \sum_{S \in \mathcal{C}} z_S & \\ \text{subject to} & \sum_{S \in \mathcal{C}} a_{iS} z_S \geq 1 & \text{for } i \in K, \\ & z_S \geq 0 & \text{for } S \in \mathcal{C}. \end{array} \right.$$

We denote the nonnegative dual variable corresponding to the set covering constraint in RMP($\mathcal{C}$) by $\boldsymbol{y} = (y_i)_{i \in K}$, and then the dual problem of RMP($\mathcal{C}$) is given as

$$\text{RMD}(\mathcal{C}): \quad \left| \begin{array}{lll} \text{minimize} & \displaystyle\sum_{i \in K} y_i & \\ \text{subject to} & \displaystyle\sum_{i \in K} a_{iS} y_i \leq 1 & \text{for } S \in \mathcal{C}, \\ & y_i \geq 0 & \text{for } i \in K. \end{array} \right.$$

Let $((z_S^*)_{S \in \mathcal{C}}, (y_i^*)_{i \in K})$ be a pair of optimal primal-dual solutions to restricted master problem. At an optimal primal solution $(z_S^*)_{S \in \mathcal{C}}$ of RMP($\mathcal{C}$), if the reduced cost for any $S \in \mathcal{S} \setminus \mathcal{C}$ is nonpositive, namely,

$$\sum_{i \in K} a_{iS} y_i^* - 1 \leq 0 \text{ for } S \in \mathcal{S} \setminus \mathcal{C},$$

then the current solution $(z_S^*)_{S \in \mathcal{C}}$ is optimal for the master problem MP as well. On the other hand, there exists $S \in \mathcal{S} \setminus \mathcal{C}$ with the positive reduced cost, then adding S to $\mathcal{C}$ brings possibly the improvement of the objective value of RMP($\mathcal{C}$). Therefore, the nonpositivity of the reduced cost implies the primal optimality condition. In the dual space, we can regard the nonpositivity of the reduced cost as the dual feasibility condition. Namely, adding S with the positive reduced cost to $\mathcal{C}$ corresponds to cutting off the current dual optimal solution $\boldsymbol{y}^*$.

Now, a main object in the column generation is to find a column $S \in \mathcal{S} \setminus \mathcal{C}$ which satisfies $\sum_{i \in K} a_{iS} y_i^* - 1 > 0$, so that we introduce a binary variable x_u which is one if a vertex u belongs to such a column and zero otherwise. Then the column generation subproblem ends up with a variant of the maximum weight stable set problem as follows.

$$\text{SP}^{\text{S}}(\boldsymbol{y}^*): \quad \left| \begin{array}{lll} \text{maximize} & \displaystyle\sum_{i \in K} y_i^* r_i - 1 & \\ \text{subject to} & x_u + x_v \leq 1 & \text{for } \{u, v\} \in E, \\ & \displaystyle\sum_{u \in V_i} x_u = r_i & \text{for } i \in K, \\ & x_u \in \{0, 1\} & \text{for } u \in V, \\ & r_i \in \{0, 1\} & \text{for } i \in K. \end{array} \right.$$

The first set of constraints and the binary constraint of x_u imply that $\boldsymbol{x} = (x_u)_{u \in V}$ is the incidence vector of a stable set, i.e., the set $\{u \in V \mid x_u = 1\}$ is a stable set. Here, an auxiliary variable r_i and the second set of constraints force a stable set constructed by $\boldsymbol{x}$ to belong to $\mathcal{S}$. Note that the binary constraint of the variable r_i is redundant owing to the second set of constraints, hence it is relaxed to $0 \leq r_i \leq 1$.

Having found $\boldsymbol{r} = (r_i)_{i \in K}$ with the positive optimal value, we add the column $\boldsymbol{r}$ to the current restricted master problem RMP($\mathcal{C}$), or equivalently add the following constraint

$$\sum_{i \in K} r_i y_i \leq 1,$$

to the problem of RMD($\mathcal{C}$). From the above discussion, outline of the column generation is described as follows.

Algorithm 1 STANDARD COLUMN GENERATION

1: Let $\mathcal{C}$ be an initial set of columns.
2: Solve RMD($\mathcal{C}$) to obtain an optimal solution $\boldsymbol{y}^*$ and the optimal value $\omega(\mathrm{RMD}(\mathcal{C}))$.
3: Solve $\mathrm{SP}^{\mathsf{S}}(\boldsymbol{y}^*)$ to obtain an optimal solution $\boldsymbol{r}$.
4: **if** $\omega(\mathrm{SP}^{\mathsf{S}}(\boldsymbol{y}^*)) \leq 0$ **then**
5: Go to Line 8.
6: **else**
7: Set $\mathcal{C} \leftarrow \mathcal{C} \cup \{\, \{i \in K \mid r_i = 1\} \,\}$ and return to Line 2.
8: Solve RMP($\mathcal{C}$) whose continuous variable z_S is replaced by the binary variable, and set an upper bound UB to its optimal value.

Note that, for any $S \in \mathcal{C}$, the reduced costs are nonpositive due to the feasibility of $\boldsymbol{y}^*$ for RMD($\mathcal{C}$). This guarantees the convergence of the algorithm, that is, the algorithm terminates within a finite number of iterations since it does not generate the column which has been already added.

3.2. **Bound-optimal column generation.** Recently, Coniglio and Tieves [2] have proposed the *bound-optimal cutting plane method*, a new paradigm in the literature of cutting planes. This method is based on an intuitive strategy of generating cuts (or columns in the dual), which yields the largest improvement of the objective value.

Adding a particular stable set to subfamily $\mathcal{C}$, we have the following problem

$$\overline{\mathrm{RMP}(\mathcal{C})}: \quad \begin{array}{lll} \text{minimize} & \displaystyle\sum_{S\in\mathcal{C}} z_S + w & \\ \text{subject to} & \displaystyle\sum_{S\in\mathcal{C}} a_{iS} z_S + r_i w \geq 1 & \text{for } i \in K, \\ & w \geq 0,\ z_S \geq 0 & \text{for } S \in \mathcal{C}, \end{array}$$

where $(r_i)_{i\in K}$ is the column corresponding to the added stable set, and we denote w be the primal variable associated with the column. In the bound-optimal strategy, we determine both the primal variables (z_S, w) and the variable r_i representing a column so as to minimize the objective value of $\overline{\mathrm{RMP}(\mathcal{C})}$, which results in the following problem

$$\mathrm{SP}^{\mathsf{B}}(\mathcal{C}): \quad \begin{array}{lll} \text{minimize} & \displaystyle\sum_{S\in\mathcal{C}} z_S + w & \\ \text{subject to} & \displaystyle\sum_{S\in\mathcal{C}} a_{iS} z_S + r_i w \geq 1 & \text{for } i \in K, \\ & x_u + x_v \leq 1 & \text{for } \{u,v\} \in E, \\ & \displaystyle\sum_{u\in V_i} x_u = r_i & \text{for } i \in K, \\ & w \geq 0,\ z_S \geq 0 & \text{for } S \in \mathcal{C}, \\ & x_u \in \{0,1\} & \text{for } u \in V, \\ & 0 \leq r_i \leq 1 & \text{for } i \in K. \end{array}$$

The differences between the standard column generation and the bound-optimal generation is that the latter unifies both steps of reoptimizing the restricted master problem and

generating a column into a single step. Namely, $\overline{\mathrm{RMP}(\mathcal{C})}$ is implicitly reoptimized in the process of solving the problem $\mathrm{SP}^{\mathsf{B}}(\mathcal{C})$.

One of the difficulties in the bound-optimal generation is due to the bilinear term $r_i w$. To linearize each bilinear term, Coniglio and Tieves [2] introduced a continuous variable λ_i to replace $r_i w$, and made use of the following McCormick inequalities [10]:

$$w - (1 - r_i) \leq \lambda_i \leq w \text{ and } 0 \leq \lambda_i \leq r_i \text{ for } i \in K. \tag{3.1}$$

As the results, the problem can be reformulated as a *Mixed Integer Linear Programming problem*, MILP for short, which enables an application of general purpose optimization solvers to the problem.

4. Our Proposed Algorithm

Coniglio and Tieves [2] have reported that the bound-optimal generation reduces the number of generated columns compared to the standard column generation. However, the bound-optimal generation has the following disadvantages: (i) the algorithm does not have the convergence property, and (ii) a heavy computational effort is required to solve $\mathrm{SP}^{\mathsf{B}}(\mathcal{C})$ at each iteration even though the reformulation to MILP, in a manner of McCormick inequalities previously mentioned, allows us to apply general purpose optimization solvers to the problem.

4.1. Convergence issue.

Among the above disadvantages, the convergence property is more crucial issue. Once the bound-optimal generation fails to improve the objective value, the method keeps on generating a column which have been already added. In [2], this situation is called *stalling iteration*. One way to avoid the stalling iteration is to restrict the reduced cost at the current dual solution $\boldsymbol{y}^*$ to being positive, which realizes to introduce the following approximated constraint

$$\sum_{i \in K} y_i^* r_i - 1 \geq \delta,$$

where $\delta (> 0)$ is a sufficiently small parameter. Then the problem we consider is given as

$$\mathrm{SP}(\mathcal{C}, \boldsymbol{y}^*, \delta): \quad \left| \begin{array}{lll} \text{minimize} & \displaystyle\sum_{S \in \mathcal{C}} z_S + w & \\ \text{subject to} & \displaystyle\sum_{S \in \mathcal{C}} a_{iS} z_S + r_i w \geq 1 & \text{for } i \in K, \\ & \displaystyle\sum_{i \in K} y_i^* r_i - 1 \geq \delta, & \\ & x_u + x_v \leq 1 & \text{for } \{u, v\} \in E, \\ & \displaystyle\sum_{u \in V_i} x_u = r_i & \text{for } i \in K, \\ & w \geq 0,\ z_S \geq 0 & \text{for } S \in \mathcal{C}, \\ & x_u \in \{0, 1\} & \text{for } u \in V, \\ & 0 \leq r_i \leq 1 & \text{for } i \in K. \end{array} \right.$$

The following lemma would give a useful information for determining the parameter δ.

Lemma 4.1. *Let $\boldsymbol{y} = (y_i)_{i \in K}$ be a feasible solution to RMD($\mathcal{C}$), and δ be a positive scalar, respectively. If the problem SP($\mathcal{C}, \boldsymbol{y}, \delta$) is infeasible, then $\sum_{i \in K} y_i/(1+\delta)$ is a lower bound on the optimal value of MP.*

Proof. If the problem is infeasible, then it holds that $\sum_{i \in K} a_{iS} y_i - 1 < \delta$ for any $S \in \mathcal{S}$, which is rewritten

$$\sum_{i \in K} a_{iS} \left(\frac{y_i}{1+\delta} \right) < 1 \text{ for any } S \in \mathcal{S}.$$

This implies that the solution $(y_i/(1+\delta))_{i \in K}$ is feasible to the dual of the master problem MP, and hence its objective value $\sum_{i \in K} y_i/(1+\delta)$ satisfies $\omega(\text{MP}) \geq \sum_{i \in K} y_i/(1+\delta)$ due to the weak duality of LP. □

From Lemma 4.1, we can obtain a lower bound θ on ω(MP) by setting

$$\theta = \omega(\text{RMD}(\mathcal{C}))/(1+\delta)$$

when the problem SP($\mathcal{C}, \boldsymbol{y}^*, \delta$) is infeasible. Obviously, a sequence $\{\theta_\ell \mid \ell = 0, 1, 2, \ldots\}$ obtained by the algorithm converges to the optimal value of MP when we take a sequence $\{\delta_\ell \mid \ell = 0, 1, 2, \ldots\}$ which satisfies $\delta_\ell \to 0$. Moreover, the above lemma together with the integrality of ω(P) provides that $\lceil \theta \rceil$ is a lower bound on ω(P). Therefore, if the problem SP($\mathcal{C}, \boldsymbol{y}^*, \delta$) is infeasible, we update δ in a simple way $\delta \leftarrow \delta/2$ until the objective value is sufficiently close to the largest lower bound on ω(P) obtained so far.

Algorithm 2 BOUND-OPTIMAL GENERATION

Let $\mathcal{C}$ be an initial set of columns and ε be a tolerance parameter, respectively.
Set counter $\ell \leftarrow 0$.
Set $\delta_0 \leftarrow 0.5$, LB $\leftarrow 1$, and $\theta_0 \leftarrow 1$.
Solve RMD($\mathcal{C}$) to obtain an optimal solution $\boldsymbol{y}^*$ and the optimal value $\omega(\text{RMD}(\mathcal{C}))$.
while $\omega(\text{RMD}(\mathcal{C})) - \text{LB} > \varepsilon$ and $\delta_\ell > \varepsilon \times 10^{-1}$ **do**
    Find any solution $\boldsymbol{r}$ of SP($\mathcal{C}, \boldsymbol{y}^*, \delta_\ell$).
    if SP($\mathcal{C}, \boldsymbol{y}^*, \delta_\ell$) is infeasible **then**
        $\theta_\ell \leftarrow \omega(\text{RMD}(\mathcal{C}))/(1+\delta_\ell)$.
        if LB $> \lceil \theta_\ell \rceil$ **then**
            LB $\leftarrow \lceil \theta_\ell \rceil$.
        Update $\delta_\ell \leftarrow \delta_\ell/2$.
    else
        $\theta_{\ell+1} \leftarrow \theta_\ell, \delta_{\ell+1} \leftarrow \delta_\ell$.
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{ \{i \in K \mid r_i = 1\} \}$.
        Solve RMD($\mathcal{C}$) to obtain an optimal solution $\boldsymbol{y}^*$ and its optimal value.
        $\ell \leftarrow \ell + 1$.
Solve RMP($\mathcal{C}$) whose continuous variable z_S is replaced by the binary variable, and set an upper bound UB to its optimal value.

4.2. **Reduction of computational effort for subproblem.** The lower bounding process (Line 8 to Line 10 in Algorithm 2) operates when the subproblem is infeasible, thus it is not necessary to solve it to optimality to this end. This prompts us to apply a heuristic algorithm to the subproblem in order to alleviate the computational burden arising from the bilinear term.

In this note, we iteratively solve the subproblem by alternately fixing one term of each bilinear term. Each iteration consists of two phases:

(i) fixing w, solve the problem with respect to $\boldsymbol{x}$ and $\boldsymbol{r}$, say the problem SP_1,
(ii) fixing $\boldsymbol{x}$ and $\boldsymbol{r}$, solve the problem with respect to $\boldsymbol{z}$ and w, say the problem SP_2, which reduces to a just LP problem.

When no improvement of the objective value of SP_1 occurs, we stop this procedure. The solution returned by the procedure is not necessarily an optimal solution, but a local optimal one. The procedure is described as follows.

Procedure 1 ALTERNATING VARIABLE FIXING

1: Set counter $i \leftarrow 0$.
2: Set $w^{(0)} \leftarrow 1.0$, $\alpha^{(0)} \leftarrow \infty$ and $stop \leftarrow false$.
3: **while** $stop = false$ **do**
4: $\quad(\boldsymbol{x}^{(i+1)}, \boldsymbol{r}^{(i+1)}) \leftarrow \mathrm{argmin}\{\,\mathrm{SP}_1(\mathcal{C}, \boldsymbol{y}^*, \delta, w^{(i)}) \mid \boldsymbol{x}, \boldsymbol{r}\,\}$ and set $\alpha^{(i+1)}$ to its optimal value.
5: $\quad(\boldsymbol{z}^{(i+1)}, w^{(i+1)}) \leftarrow \mathrm{argmin}\{\,\mathrm{SP}_2(\mathcal{C}, \boldsymbol{y}^*, \delta, \boldsymbol{x}^{(i+1)}, \boldsymbol{r}^{(i+1)}) \mid \boldsymbol{z}, w\,\}$.
6: $\quad$**if** $\alpha^{(i)} - \alpha^{(i+1)} \leq \varepsilon$ **then**
7: $\quad\quad stop \leftarrow true$.
8: $\quad i \leftarrow i + 1$.

Note that the problem SP_2 corresponds to $\mathrm{RMP}(\mathcal{C})$, hence its dual optimal solution $\boldsymbol{y}^*$ can be derived from an optimal solution $(\boldsymbol{z}, w)$ of SP_2. This implies that we need not reoptimize the restricted master problem, at Line 15 in Algorithm 2.

5. COMPUTATIONAL EXPERIMENTS

5.1. **Computational environment and solved instances.** In this section, we present experimental results to evaluate our bound-optimal generation in terms of the quality of solution obtained, the number of generated columns, and the computation time. To clarify the performance of our algorithm, we compare our bound-optimal generation (Algorithm 2) with the standard column generation (Algorithm 1, denoted by STD). We execute Algorithm 2 in two ways, one of which solves the subproblem at Line 6 by using McCormick inequalities (3.1), and the other does the subproblem by using Procedure 1. We refer to the former as BOG-MC, and the latter as BOG-ALT, respectively. To compare BOGs with STD in a fair setting, we use $\omega(\mathrm{RMD}(\mathcal{C})) - \mathrm{LB} \leq \varepsilon$ as one of the stopping conditions of STD, where LB is given as $\lceil \omega(\mathrm{RMD}(\mathcal{C}))/(1 + \mathrm{SP}^{\mathrm{S}}(\boldsymbol{y}^*)) \rceil$ in the case of STD. For each algorithm, we set the tolerance parameter $\varepsilon = 10^{-4}$ and collect one singleton for each component V_i to make $\mathcal{C}$ as an initial set of columns.

We executed experiments on a Mac OSX computer with Intel Core i5@3.4GHz processor and 8GB of memory. The algorithms are implemented in Python 3.6.1, calling Gurobi Optimizer 7.5.1 as both LP and MILP solvers. For each algorithm, we impose a time limit of 3600 seconds. In a case where each algorithm reaches time limit, we collect all columns generated until then and solve the problem to obtain an upper bound.

We solved a set of 50 instances consisting of the National Science Foundation Network (called nsf), which is available from `http://www2.ic.uff.br/~celso/grupo/pcp.htm`.

5.2. **Experimental results.** The description of the nsf instances and the computational results for STD, BOG-MC and BOG-ALT on the instances are summarized in Table 1, where the columns labeled 'UB' and 'LB' represent the upper and lower bounds finally obtained by each algorithm, the column labeled 'Cols' represents the number of columns generated, and the column labeled 'Time' represents the computation time in seconds, respectively.

Form Table 1, we can observe that BOGs generate the smaller number of columns than STD in most of the instances, whereas the BOGs require a long computation time. In particular, BOG-MC does not converge within the time limit for 11 instances out of whole of the instances. As for the quality of solution obtained, BOG-ALT outputs better upper bound solutions than both BOG-MC and STD in all instances. It is notable that BOG-ALT solves the instances except the instance p1.0s3 to optimality at the root node.

For the instances not solved at the root node, we have tackled them with a branch-and-price approach which is the column generation combined with branch-and-bound. The computational results for the branch-and-price approaches (with BOG-ALT and STD as the column generation phase) on unsolved instance p1.0s3 are provided in Table 2, where the columns labeled 'Node', '$\mathrm{UB_{BP}}$', '$\mathrm{LB_{BP}}$', '$\mathrm{Cols_{BP}}$' and '$\mathrm{Time_{BP}}$' represent the number of branch-and-bound tree nodes explored, the upper and lower bounds finally obtained, the number of columns generated (including columns generated at the root node), and the computation time in seconds (including the computation time at the root node), throughout the branch-and-price process. From Table 2, we observe that the branch-and-price with BOG-ALT outputs an optimal solution after exploring five branch-and-bound tree nodes. On the other hand, the branch-and-price with STD has not converged within the time limit, and we have similar results for the other instances unsolved at the root node.

TABLE 1. Computational results.

Instance				STD				BOG-MC				BOG-ALT			
Name	n	m	k	UB	LB	Cols	Time	UB	LB	Cols	Time	UB	LB	Cols	Time
p0.1s1	16	16	12	2	2	2	<0.1	2	2	6	<0.1	2	2	3	<0.1
p0.1s2	22	27	16	2	2	11	<0.1	2	2	4	<0.1	2	2	4	<0.1
p0.1s3	29	40	23	3	3	18	<0.1	3	3	5	<0.1	3	3	4	<0.1
p0.1s4	38	67	30	4	3	47	<0.1	3	3	9	<0.1	3	3	7	<0.1
p0.1s5	27	30	20	2	2	2	<0.1	2	2	11	<0.1	2	2	6	<0.1
p0.2s1	37	69	31	4	4	31	<0.1	4	4	16	0.1	4	4	16	0.1
p0.2s2	52	128	44	5	4	51	<0.1	4	4	33	0.7	4	4	23	0.3
p0.2s3	51	130	40	4	4	81	0.1	4	4	35	1.2	4	4	29	0.7
p0.2s4	57	176	40	4	4	19	<0.1	4	4	11	0.1	4	4	10	<0.1
p0.2s5	66	242	44	5	4	70	0.1	4	4	49	8.6	4	4	46	3.5
p0.3s1	63	220	49	5	5	27	<0.1	5	5	25	0.6	5	5	22	0.3
p0.3s2	87	452	64	6	5	93	0.2	6	5	43	7.8	5	5	47	1.2
p0.3s3	80	366	58	7	6	44	0.1	6	6	30	1.4	6	6	35	1.0
p0.3s4	80	397	59	7	6	71	0.2	6	6	38	2.3	6	6	32	0.9
p0.3s5	85	363	63	6	5	116	0.5	5	5	63	18.8	5	5	62	6.8
p0.4s1	91	527	66	8	6	68	0.2	7	6	39	5.9	6	6	40	1.9
p0.4s2	112	739	82	9	7	89	0.3	7	7	49	8.6	7	7	48	3.3
p0.4s3	101	606	73	8	6	69	0.2	6	6	40	2.3	6	6	40	1.5
p0.4s4	99	559	76	8	7	77	0.2	7	7	37	4.6	7	7	44	2.1
p0.4s5	112	741	80	8	6	153	0.8	7	6	82	208.4	6	6	70	12.0
p0.5s1	124	999	87	10	8	79	0.3	8	8	52	19.5	8	8	48	3.1
p0.5s2	130	1017	99	10	8	96	0.4	9	8	54	19.8	8	8	51	5.2
p0.5s3	122	848	92	9	7	131	0.6	8	7	72	59.3	7	7	78	14.1
p0.5s4	118	803	89	8	7	108	0.4	8	7	54	22.3	7	7	54	4.2
p0.5s5	132	1071	93	9	7	129	0.6	8	7	74	158.2	7	7	78	18.5
p0.6s1	149	1475	107	11	9	100	0.5	10	9	58	78.7	9	9	52	2.8
p0.6s2	154	1409	113	12	9	94	0.4	10	9	52	42.9	9	9	58	5.2
p0.6s3	153	1371	112	12	9	109	0.8	9	9	64	38.0	9	9	65	14.0
p0.6s4	161	1613	113	11	9	131	1.0	9	9	77	196.8	9	9	74	37.3
p0.6s5	139	1184	103	11	9	72	0.3	10	9	47	40.3	9	9	42	3.3
p0.7s1	180	2117	124	13	10	166	6.3	11	10	95	900.4	10	10	94	143.3
p0.7s2	202	2670	144	14	11	122	0.9	11	11	68	491.9	11	11	74	18.5
p0.7s3	177	1925	131	13	10	114	0.7	10	10	79	122.0	10	10	59	9.0
p0.7s4	187	2151	135	14	11	138	1.5	11	11	80	425.0	11	11	71	34.5
p0.7s5	159	1606	118	11	9	148	1.0	9	9	88	1447.4	9	9	78	32.5
p0.8s1	201	2549	147	15	11	144	2.7	12	1	82	>3600	11	11	107	35.8
p0.8s2	221	3115	153	15	11	177	27.8	13	1	73	>3600	11	11	119	163.7
p0.8s3	208	2692	147	15	11	127	2.0	11	11	62	142.4	11	11	64	10.8
p0.8s4	209	2821	147	15	11	157	7.4	11	1	82	>3600	11	11	104	299.4
p0.8s5	184	2014	139	14	11	96	0.5	11	11	53	70.2	11	11	60	8.5
p0.9s1	216	2921	161	16	12	158	4.1	12	1	79	>3600	12	12	93	54.1
p0.9s2	231	3324	167	16	12	152	2.0	12	12	97	2598.0	12	12	70	25.0
p0.9s3	217	2849	166	16	12	144	1.5	13	12	72	511.4	12	12	82	23.9
p0.9s4	226	3118	165	16	12	212	402.8	12	8	101	>3600	12	12	144	1024.7
p0.9s5	234	3379	169	16	12	189	85.5	12	1	94	>3600	12	12	91	226.4
p1.0s1	250	3948	182	18	13	206	153.1	14	1	71	>3600	13	13	112	117.8
p1.0s2	251	3973	182	18	13	235	2294.1	14	1	73	>3600	13	13	129	1172.7
p1.0s3	238	3419	182	18	13	188	73.4	14	1	75	>3600	14	13	107	1080.0
p1.0s4	257	4261	182	17	13	202	8.1	14	1	73	>3600	13	13	134	846.1
p1.0s5	248	3827	182	18	13	187	74.7	14	1	63	>3600	13	13	80	19.1

TABLE 2. Computational results of the branch-and-price.

Instance				STD					BOG-ALT				
Name	n	m	k	Node	UB_{BP}	LB_{BP}	$Cols_{BP}$	$Time_{BP}$	Node	UB_{BP}	LB_{BP}	$Cols_{BP}$	$Time_{BP}$
p1.0s3	238	3419	182	11	18	13	215	>3600	2	13	13	148	1239.9

References

[1] M. Campelo, R. Correa, and Y. Frota. Cliques, holes and the vertex coloring polytope. *Information Processing Letters*, 89(4):159–164, 2004.

[2] S. Coniglio and M. Tieves. On the generation of cutting planes which maximize the bound improvement. In *International Symposium on Experimental Algorithms*, pages 97–109. Springer, 2015.

[3] M. Demange, T. Ekim, B. Ries, and C. Tanasescu. On some applications of the selective graph coloring problem. *European Journal of Operational Research*, 240:307–314, 2015.

[4] M. Demange, J. Monnot, P. Pop, and B. Ries. On the complexity of the selective graph coloring problem in some special classes of graphs. *Theoretical Computer Science*, 540:89–102, 2014.

[5] Y. Frota, N. Maculan, T. F. Noronha, and C. C. Ribeiro. A branch-and-cut algorithm for partition coloring. *Networks*, 55(3):194–204, 2010.

[6] F. Furini and E. Malaguti. Exact weighted vertex coloring via branch-and-price. *Discrete Optimization*, 9(2):130–136, 2012.

[7] F. Furini, E. Malaguti, and A. Santini. Exact and heuristic algorithms for the partition coloring problem.

[8] E. A. Hoshino, Y. A. Frota, and C. C. De Souza. A branch-and-price approach for the partition coloring problem. *Operations Research Letters*, 39(2):132–137, 2011.

[9] 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*, page 1. Citeseer, 2000.

[10] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[11] A. Mehrotra and M. A. Trick. A branch-and-price approach for graph multi-coloring. In *Extending the horizons: Advances in computing, optimization, and decision technologies*, pages 15–29. Springer, 2007.

[12] B. Yuceoglu, G. Sahin, and S. P. v. Hoesel. A column generation based algorithm for the robust graph coloring problem. *Discrete Applied Mathematics*, 217(2):340–352, 2017.

(Y. Izunaga) Faculty of Business Sciences, University of Tsukuba, Otsuka 3-29-1, Bunkyo-ku, Tokyo 112-0012, Japan.

E-mail address: izunaga@gssm.otsuka.tsukuba.ac.jp

(K. Sato) Signaling and Transport Information Technology Division, Railway Technical Research Institute, Hikari-cho 2-8-38, Kokubunji-shi, Tokyo 185-8540, Japan.

E-mail address, K. Sato: sato.keisuke.49@rtri.or.jp