Set covering and packing formulations of graph coloring: algorithms and first polyhedral results

P. Hansen*
M. Labbé†
D. Schindl‡

September 29, 2005

Abstract

We consider two (0,1)-linear programming formulations of the graph (vertex-) coloring problem, in which variables are associated to stable sets of the input graph. The first one is a set covering formulation, where the set of vertices has to be covered by a minimum number of stable sets. The second is a set packing formulation, in which constraints express that two stable sets cannot have a common vertex, and large stable sets are preferred in the objective function. We identify facets with small coefficients for the polytopes associated with both formulations. We show by computational experiments that both formulations are about equally efficient when used in a branch-and-price algorithm. Next we propose some preprocessing, and show that it can substantially speed up the algorithm, if it is applied at each node of the enumeration tree. Finally we describe a cutting plane procedure for the set covering formulation, which often reduces the size of the enumeration tree.

Keywords: Graph coloring, stable sets, facets, branch-cut-and-price algorithm

1 Introduction

The *graph (vertex-)coloring problem* (GC) consists, given a graph $G = (V, E)$, in assigning a color to each vertex of G such that any two adjacent vertices receive different colors, and the total number of colors used is minimized. The minimum

*HEC Montreal, GERAD et Méthodes quantitatives de gestion, Pierre.Hansen@gerad.ca
†Université Libre de Bruxelles, Dép. d'Informatique, mlabbe@ulb.ac.be
‡HEC Montreal, GERAD, david.schindl@a3.epfl.ch

number of colors necessary to color G is called the *chromatic number* of G and denoted by $\chi(G)$. Problem GC is one of the most important in graph theory, it presents many applications (e.g. in timetabling, scheduling and frequency assignment) and has been extensively studied, see e.g. [Sch03].

For problem GC, several authors proposed exact algorithms, based on integer programming formulations. Mehrotra and Trick ([MT96]) use a formulation of the problem using variables associated to maximal stable sets of the graph. Since such a formulation may contain a huge number of variables, Mehrotra and Trick use a column generation approach and special branching rules. In [CMaZ02], Coll, Marenco, Méndez Díaz and Zabala consider a formulation with a polynomial number of variables corresponding to color assignments to vertices. The linear relaxation of this formulation is extremely weak but the authors identify powerful families of valid inequalities which allow to reduce significantly the integrality gap.

In Section 2, we present three (a covering, a packing and a partitioning) integer linear programming formulations for the graph coloring problem. They all involve an exponential number of variables. The bounds obtained by their respective linear relaxations are shown to be equivalent in Section 3. In Section 4, we focus on the set covering formulation and all facets among the inequalities of the initial formulation are characterized. We also provide conditions related to less trivial inequalities. In Section 5, we study the set packing formulation, essentially by considering facets induced by maximal cliques in the associated conflict graph. Similarly as in Section 4, we again characterize all facets among the inequalities of the initial formulation. We then extend this result to a larger class, i.e. facets associated with some but not all maximal cliques. In Section 6, we present computational results on branch-and-price algorithms obtained with those formulations. A preprocessing and a cutting plane procedure for the covering formulation are described and tested. Brief conclusions are stated in Section 7.

2 Notations and formulations

Let $G = (V, E)$ be a graph. Two vertices v and w of V such that $(v, w) \in E$ are said to be *adjacent*. For $v \in V$, the *neighborhood* of v is the set $N(v) = \{w \in V : (v, w) \in E\}$. The neighborhood of a subset $V' \subseteq V$ is $N(V') = \bigcup_{v \in V'} N(v) - V'$. For $v \in V$, the *anti-neighborhood* of v is the set $AN(v) = V - \{N(v) \cup v\}$. The anti-neighborhood of a subset $V' \subseteq V$ is $AN(V') = \bigcap_{v \in V'} AN(v)$. The *complementary* graph $\overline{G} = (V, \overline{E})$ of G is the graph in which two vertices are adjacent if and only if they are not in G. For $V' \subseteq V$, the *subgraph* of G *induced* by V' is $G[V'] = (V', E \cap (V' \times V'))$. A *stable set* of G is a set of vertices $S \subseteq V$, such that no two vertices of S are adjacent. The maximum size of a stable set in G is called the *stability number* of G and denoted by $\alpha(G)$. A *clique* is a set of vertices $C \subseteq V$, such that each vertex in C is adjacent to each other vertex in C. It can also be seen as a stable set in $\overline{G}$. The maximum size of a clique in G is called the *clique number* of G and is denoted by $\omega(G)$.

Let $\overline{\chi}$ be an upper bound on the chromatic number of G. A standard integer linear programming formulation for the graph coloring problem is given below, where $y_j = 1$ if color j is used and $x_{vj} = 1$ if vertex v receives color j. Constraints (6) impose that each vertex is in exactly one stable set, constraints (3) force two adjacent vertices to receive different colors, and the objective value is equal to the number of colors used. This is the formulation used in [CMaZ02].

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{\overline{\chi}} y_j & & & (1)\\
s.t. \quad & \sum_{j=1}^{\overline{\chi}} x_{vj} = 1 & \forall\, v \in V & \qquad (GC(St)) & (2)\\
& x_{vj} + x_{wj} \leq y_j & \forall\, [v,w] \in E,\ j \in \{1,\ldots,\overline{\chi}\} & & (3)\\
& x_{vj}, y_j \in \{0,1\} & \forall\, v \in V,\ j \in \{1,\ldots,\overline{\chi}\} & & (4)
\end{aligned}
$$

We then consider the following set partitioning formulation.

$$
\begin{aligned}
\min \quad & \sum_{S\in\mathcal{S}} x_S & & & (5)\\
s.t. \quad & \sum_{S\in\mathcal{S}:v\in S} x_S = 1 & \forall v \in V & \qquad (GC(Part)) & (6)\\
& x_S \in \{0,1\} & \forall S \in \mathcal{S} & & (7)
\end{aligned}
$$

in which $\mathcal{S}$ is the set of all stable sets of G, and $x_S = 1$ if the stable set S corresponds to a color class. Denote by $\mathcal{S}_2 = \{S \in \mathcal{S} : |S| \geq 2\}$.
One can replace variables x_S such that $|S| = |\{v\}| = 1$, by $1 - \sum_{\{S\in\mathcal{S}_2:v\in S\}} x_S$. This substitution guarantees the satisfaction of constraints (6), but the fact that $1 - \sum_{S\in\mathcal{S}_2} x_S = x_{\{v\}} \geq 0\ \forall\ v$ brings the inequalities $\sum_{S\in\mathcal{S}_2} x_S \leq 1$. After having transformed the objective function accordingly, we get the following formulation.

$$
\begin{aligned}
\max \quad & \sum_{S\in\mathcal{S}_2} (|S|-1)x_S & & & (8)\\
s.t. \quad & \sum_{S\in\mathcal{S}_2:v\in S} x_S \leq 1 & \forall v \in V & \qquad (GC(Pack)) & (9)\\
& x_S \in \{0,1\} & \forall S \in \mathcal{S}_2 & & (10)
\end{aligned}
$$

Another formulation can be obtained by observing that partitioning V is not necessary, but only covering it with a minimum number of stable sets suffices to solve the graph coloring problem. Moreover, in a covering, a stable set which is not inclusionwise maximal can be replaced by such a maximal one. Hence we only need to take into account variables corresponding to maximal stable sets. Let $\mathcal{S}_{max}$ be those sets, we then have the following set covering formulation, which is the one used in [MT96].

$$\min \quad \sum_{S\in\mathcal{S}_{max}} x_S \tag{11}$$
$$s.t. \quad \sum_{S\in\mathcal{S}_{max}:v\in S} x_S \geq 1 \quad \forall v \in V \qquad (GC(Cov)) \tag{12}$$
$$x_S \in \{0,1\} \qquad \forall S \in \mathcal{S}_{max} \tag{13}$$

3 Comparison of the linear relaxations

As mentioned in the introduction, the linear relaxation $GC^l(St)$ of the standard formulation provides a very bad lower bound on $\chi(G)$. Indeed, consider the fractional solution given by $x_{ij} = \frac{1}{2}, y_j = 1$ for all i if $j = 1, 2$, and $x_{ij} = y_j = 0$ for all i if $j \geq 3$. This solution satisfies constraints (3) whatever is the instance graph, so the optimal value $z^{*l}(St)$ of $GC^l(St)$ is at most 2 (it is 1 if the graph has no edges). Since $\chi(G)$ may be arbitrarily large, this lower bound is useless.

Consider the linear relaxations $GC^l(Part), GC^l(Pack)$ and $GC^l(Cov)$ of the above formulations, denote by $x^*(Part)$, $x^*(Pack)$ and $x^*(Cov)$ their optimal solutions and $z^{*l}(Part)$, $z^{*l}(Pack)$ and $z^{*l}(Cov)$ their optimal values, respectively. Since $GC(Part)$ is obtained by applying Dantzig-Wolfe decomposition to constraints (3) and (4) (see e.g. [Sch04]), it immediately follows that $z^{*l}(St) \leq z^{*l}(Part)$. The following proposition shows that the bounds obtained by the three linear relaxations $GC^l(Part), GC^l(Pack)$ and $GC^l(Cov)$ are in fact of the same quality.

Proposition 1 $z^{*l}(Cov) = z^{*l}(Part) = |V| - z^{*l}(Pack)$.

Proof. A development similar to the one above leading from $GC(Part)$ to $GC(Pack)$ permits to prove that problems $GC^l(Pack)$ and $GC^l(Part)$ are equivalent. Furthermore, we have

$$z^{*l}(Part) = \sum_{S\in\mathcal{S}} x^*_S(Part) = \sum_{S\in\mathcal{S}} x^*_S(Part) + (|V| - \sum_{S\in\mathcal{S}} |S| x^*_S(Part))$$

$$= |V| - \sum_{S\in\mathcal{S}} (|S|-1) x^*_S(Part) = |V| - \sum_{S\in\mathcal{S}_2} (|S|-1) x^*_S(Pack) = |V| - z^{*l}(Pack).$$

Hence it remains only to show that $z^{*l}(Cov) = z^{*l}(Part)$.

From a feasible solution $x(Part)$ of $GC^l(Part)$, one can always construct a feasible solution of same value $x(Cov)$ of $GC^l(Cov)$ by applying the following steps:

1. set $x_S(Cov) = 0 \ \forall S \in \mathcal{S}_{max}$;

2. for each $S \in \mathcal{S}$ such that $x_S(Part) > 0$, choose a set $S' \in \mathcal{S}_{max}$ containing S, and set $x_{S'}(Cov) = x_{S'}(Cov) + x_S(Part)$.

Since the objective function coefficients are all equal to one in both formulations, this proves that $z^{*l}(Cov) \leq z^{*l}(Part)$.

Now given a solution $x(Cov)$ of $GC^l(Cov)$, apply the following steps to obtain a solution $x(Part)$ of $GC^l(Part)$, with objective value equal to the value of $x(Cov)$.

1. set $x_S(Part) = x_S(Cov)$ if $S \in \mathcal{S}_{max}$ and $x_S(Part) = 0$ if $S \in (\mathcal{S} - \mathcal{S}_{max})$;
2. for each $v \in V$ define $a(v) = \sum\limits_{S \ni v} x_S(Part)$;
3. for each $S \in \mathcal{S}_{max}$ such that $x_S(Cov) > 0$, do:
   (a) partition S into classes $S_1, \ldots, S_q$, where v_1 and v_2 belong to the same class S_i if $a(v_1) = a(v_2) =: a(S_i)$, and order them by decreasing order of $a(S_i)$;
   (b) set $S_{temp} = S$
   (c) for $i = 1$ to q, do:
      i. set $\delta_x = \min(a(S_i) - 1, x_{S_{temp}}(Part))$;
      ii. set $x_{S_{temp}}(Part) = x_{S_{temp}}(Part) - \delta_x$;
      iii. set $S_{temp} = S_{temp} - S_i$;
      iv. set $x_{S_{temp}}(Part) = \delta_x$.

At the beginning of the algorithm, $x(Part)$ has the same value as $x(Cov)$, but may not be a feasible solution of $GC^l(Part)$, since one or more inequalities of $GC^l(Cov)$ may not be satisfied with equality. Each time $x(Part)$ is modified (steps 3(c)ii and 3(c)iv) the values of $a(S_i)$ change for some i, and the specific choices for δ_x ensure that they finally all become equal to one. Moreover, the objective value $\sum_{S \in \mathcal{S}} x_S(Part)$ at the beginning of each loop in 3c is the same as at the end of it. Hence the algorithm produces a feasible solution $x(Part)$ of $GC^l(Part)$ with same value as $x(Cov)$, which shows that $z^{*l}(Cov) \geq z^{*l}(Part)$ and permits to conclude. ■

The optimal solutions $x^*(Cov), x^*(Part)$ and $x^*(Pack)$ correspond to colorings of G with fractional colors, such that each pair of adjacent vertices are colored with disjoint sets of colors, and such that the sum of the fractions of colors corresponding to a given vertex is at least 1. The values $z^{*l}(Cov) = z^{*l}(Part) = |V| - z^{*l}(Pack)$ give a lower bound of good quality on $\chi(G)$, better known as the *fractional chromatic number* of G and denoted by $\chi_f(G)$. See [Sch97] for some of its properties.

4 Polyhedral results for the set covering formulation

Let $Cov(\mathcal{S}_{max}, V)$ be the set of feasible solutions to $GC(Cov)$; i.e. $Cov(\mathcal{S}_{max}, V) = \{x \in \{0,1\}^{|\mathcal{S}_{max}|} : (12)\}$ and denote its convex hull by $Conv(Cov(\mathcal{S}_{max}, V))$. The dimension of $Conv(Cov(\mathcal{S}_{max}, V))$ is $|\mathcal{S}_{max}|$ if and only if $\mathcal{S}_{max} \backslash \{S\}$ is a cover of V, for any $S \in \mathcal{S}_{max}$. This amounts to say that each vertex v of V belongs to at least two maximal stable sets of G, which is achieved if and only if for all $v \in V$, $V \backslash N(v)$ is not a stable set. In the opposite case, we can remove $S = V \backslash N(v)$ from G and solve GC on the reduced graph induced by $N(v)$. The optimal coloring of G would then be obtained by adding S with a new color to the optimal coloring obtained for $N(v)$. We will thus assume in this section that each vertex belongs to at least two maximal stable sets, and hence that $Conv(Cov(\mathcal{S}_{max}, V))$ is full-dimensional.

In [CS89, Sas89], the useful concept of *bipartite incidence graph* is defined to study set covering polytopes. In the special case of the graph coloring problem, the bipartite incidence graph $B(\mathcal{S}_{max}, V, E)$ is the bipartite graph with node sets $\mathcal{S}_{max}$, V and with edge set $E = \{(S, v) \in \mathcal{S}_{max} \times V : v \in S\}$. A subset S' of $\mathcal{S}_{max}$ such that each node of V has at least one neighbor in S' will be called a *cover* of V. It is easy to see that the covers of V are exactly the solutions of $Cov(\mathcal{S}_{max}, V)$. The cardinality of a minimum cover of V is called the *covering number* of V and will be denoted by $\beta(V)$. To avoid confusion, we will denote by $N_G(v)$ the neighborhood (set of vertices which are adjacent to v) of a vertex v if we refer to the graph G and $N_B(v)$ if we refer to the bipartite incidence graph B.

The following result was proved in [CS89] for the general set covering polytope. For the ease of exposition, we present it in terms of the bipartite graph $B(\mathcal{S}_{max}, V, E)$ associated to $Cov(\mathcal{S}_{max}, V)$.

Proposition 2 *Let $\mathcal{S}^1 \subseteq \mathcal{S}_{max}$ and $V^1 = \{v \in V : N_B(v) \subseteq \mathcal{S}^1\}$. Assume that $\sum_{S \in \mathcal{S}^1} x_S \geq \beta$ defines a facet of $Conv(Cov(\mathcal{S}^1, V^1))$. Then it defines a facet of $Conv(Cov(\mathcal{S}_{max}, V))$ if and only if for every $S \notin \mathcal{S}^1$,*

$$\beta(V^1 \cup V^2) = \beta(V^1)$$

where $V^2 = \{v \in V : N_B(v) \cap \mathcal{S}^1 \neq \emptyset \text{ and } N_B(v) \backslash \mathcal{S}^1 = \{S\}\}$.

Notice that since each vertex of G belongs to at least two maximal stable sets, $\{v \in V : N_B(v) = \{S\}\} = \emptyset$. In the next section this result is used to give a necessary and sufficient condition for an inequality of the form $\sum_{S \in \mathcal{S}'} x_S \geq 1$ ($\mathcal{S}' \subseteq \mathcal{S}_{max}$) to define a facet.

4.1 All facets with right hand side equal to 1

Consider the set covering formulation $GC(Cov)$ of the graph coloring problem and $Cov(\mathcal{S}_{max}, V)$ its set of feasible solutions. For a pair $v \in V$ and $w \in V$ of vertices, we say that v *dominates* w if $N_G(w) \subset N_G(v)$ (it follows that v and w are not adjacent). Notice that in this case, we have $N_B(v) \subset N_B(w)$. We will denote by $\mathcal{S}_v$ the set of all maximal (inclusionwise) stable sets containing v.

Proposition 3 *Let v be a vertex of V. Then the inequality*

$$\sum_{S \in \mathcal{S}_v} x_S \geq 1$$

defines a facet of $Conv(Cov(\mathcal{S}_{max}, V))$ if and only if v is not dominated.

Proof. Let S be a maximal stable set not containing vertex v and consider the bipartite incidence subgraph $B'(\mathcal{S}_v \cup \{S\}, V', E')$ where $V' = \{w \in V : N_B(w) \subseteq \mathcal{S}_v \cup \{S\}\}$ and $\{S', v\} \in E'$ iff $v \in S'$. Since v is not dominated then for every $w \in V'$ either $N_{B'}(w) = N_{B'}(v) = \mathcal{S}_v$ or $S \in N_{B'}(w)$ (i.e. $N_{B'}(w)$ cannot be strictly included in $N_{B'}(v)$). We can thus partition V' into two subsets: $V^1 = \{w \in V' : N_B(w) = N_B(v)\}$ and $V^2 = \{w \in V' : S \in N_{B'}(w)\}$. Note that $v \in V^1$ and $N_{B'}(w) \cap \mathcal{S}^1 \neq \emptyset \; \forall \, w \in V^2$, see Figure 1.

Clearly, $\sum_{S \in \mathcal{S}_v} x_S \geq 1$ defines a facet of $Conv(Cov(\mathcal{S}_v, V^1))$. In order to apply Proposition 2, we have to show that $\beta(V^1 \cup V^2) = \beta(V^1) = 1$. Consider $v' \in V^1$ and $w \in V^2$. Since $N_{B'}(w) \cap \mathcal{S}_v \neq \emptyset$, v' and w occur in a common stable set and hence are not adjacent in G. Moreover, w cannot be adjacent to another vertex w' of V^2, since both w' and w belong to the stable set S. Hence V^2 is stable, and by a similar argument, so is V^1. Thus $V^1 \cup V^2$ is stable and must be contained in a stable set from $\mathcal{S}_v$. So $\beta(V^1 \cup V^2) = 1$. ■

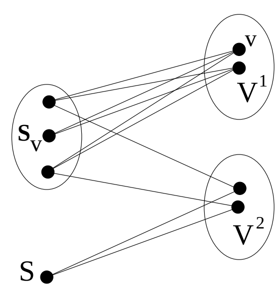

Figure 1: The bipartite incidence graph of the proof of Proposition 3.

A dominated vertex v in G is irrelevant for $\chi(G)$, since in any optimal coloring, it can have the same color as the dominating vertex. So one can remove all dominated vertices from G without decreasing $\chi(G)$. If this preprocessing is done, all inequalities (12) are facet defining.

Moreover, if no vertex of G is dominated, those facets, which will be called *vertex cover facets*, are the only one defining inequalities with right hand side equal to 1. Indeed, consider such an inequality $\sum_{S\in\mathcal{S}'} x_S \geq 1$ with $\mathcal{S}' \neq \mathcal{S}_v, v \in V$. To have a chance to define a facet, $\mathcal{S}'$ cannot have any set $\mathcal{S}_v = \{S \in \mathcal{S} : S \ni v\}$ as a subset, since the inequality $\sum_{S\in\mathcal{S}_v} x_S \geq 1$ itself defines a facet. But then the solution

$$x_S = \begin{cases} 0 & if\ S \in \mathcal{S}' \\ 1 & otherwise \end{cases}$$

covers all vertices of V, but violates the inequality $\sum_{S\in\mathcal{S}'} x_S \geq 1$.

4.2 Facet defining inequalities with right hand side larger than 1

A graph G is called *k-critical* if, for any $v \in V$, $\chi(G[V\backslash\{v\}]) = \chi(G) - 1 = k - 1$. We will say that G is $\chi - critical$ if it is $\chi(G) - critical$. We use another auxiliary graph $G^* = (\mathcal{S}_{max}, E^*)$, where $E^* = \{(S, S') : \beta(V\backslash(S \cap S')) = \beta(V) - 1\}$. Notice that $\beta(V\backslash(S\cap S'))$ is either $\beta(V)$ or $\beta(V)-1$. This last case occurs only when $S\cap S'$ intersects all χ-critical subgraphs of G. We first provide another set of inequalities with all coefficients equal to one. In [Sas89], graph G^* is used to give a sufficient condition for a class of such inequalities to be facet defining. We translate it here in our terms.

Lemma 1 *If $Conv(Cov(\mathcal{S}_{max}, V))$ is full dimensional and if G^* is connected, the inequality $\sum_{S\in\mathcal{S}_{max}} x_S \geq \chi(G)$ is facet defining.*

This lemma can be extended to the following necessary and sufficient condition.

Proposition 4 *Let G be a χ-critical graph. The inequality*

$$\sum_{S\in\mathcal{S}_{max}} x_S \geq \chi(G)$$

is facet defining if and only if the complementary graph $\overline{G}$ is connected.

Proof. *Necessity.* Denote by V' a subset of V such that $\overline{G}[V']$ is not connected to $\overline{G}[V\backslash V']$. Since no stable set in G can intersect both V' and $V\backslash V'$, we can partition $\mathcal{S}_{max}$ in two parts $\mathcal{S}'$ and $\mathcal{S}_{max}\backslash\mathcal{S}'$. Then both inequalities $\sum\limits_{S\in\mathcal{S}'} x_S \geq \chi(G[V'])$ and $\sum\limits_{S\in\mathcal{S}_{max}\backslash\mathcal{S}'} x_S \geq \chi(G[V\backslash V'])$ are valid, and their sum gives $\sum\limits_{S\in\mathcal{S}_{max}} x_S \geq \chi(G)$, which is thus not facet defining.

Sufficiency. The complementary graph $\overline{G}$ is connected iff there is a path from any vertex v to w in $\overline{G}$, or iff there is a sequence of cliques $C_1, C_2, \ldots, C_k$ in $\overline{G}$, such that $C_i \cap C_{i+1} \neq \emptyset$ for all $i \in \{1, \ldots, k-1\}$, $v \in C_1$ and $w \in C_k$. This is equivalent to saying that there is a sequence of stable sets in G with the same property, which holds also if we restrict to maximal stable sets. Moreover, since G is critical and from the above remark, for any $S_1, S_2 \in \mathcal{S}_{max}$ such that $S_1 \cap S_2 \neq \emptyset$, $\chi(G[V \backslash (S_1 \cap S_2)]) = \chi(G) - 1$. So $E^* = \{(S_1, S_2) : S_1 \cap S_2 \neq \emptyset\}$ and G^* is connected. We can now apply Lemma 1, which permits to conclude. ■

Facets with right hand side larger than 1 may be obtained using Chvátal's well-known procedure (see [Chv73]):

1. Sum a subset $Q' \subseteq Q$ of inequalities from the initial set (12);
2. divide the resulting inequality by a positive number k;
3. round up all coefficients and the right hand side to the nearest integer.

Chvátal called the *elementary closure* of a set of inequalities Q the set of inequalities which can be obtained from Q with one iteration of this procedure.

Applying this for any $V' \subseteq V$ and $k > 0$ gives a valid inequality for $Conv(Cov(\mathcal{S}_{max}, V))$. But specific choices of V' and k may lead to inequalities which are not dominated by the initial ones and even facets of $Conv(Cov(\mathcal{S}_{max}, V))$. We will see in the following that inequalities belonging to the elementary closure of the set (12) have a natural interpretation.

Example. Consider the graph G of Figure 2 with chromatic number 4 and clique number 3.

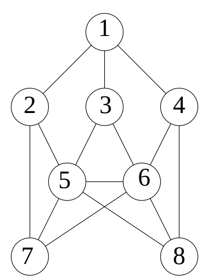

Figure 2: A graph G such that $\chi(G) = 4$ and $\omega(G) = 3$.

The set of all maximal stable sets of G is:

$$\mathcal{S}_{max} = \{\{1,5\}, \{1,6\}, \{1,7,8\}, \{2,3,4\}, \{2,3,8\}, \{2,6\}, \{3,4,7\}, \{3,7,8\}, \{4,5\}\}.$$

The inequality

$$x(1,7,8) + x(2,3,4) + x(2,3,8) + x(3,4,7) + 2x(3,7,8) \geq 2$$

defines a facet of $Conv(Cov(\mathcal{S}_{max}, V))$ which can be obtained by applying the rounding procedure to the set of inequalities

$$\begin{array}{lllllll}
 & x(2,3,4) & + x(2,3,8) & + x(3,4,7) & + x(3,7,8) & \geq & 1 \\
x(1,7,8) & & & + x(3,4,7) & + x(3,7,8) & \geq & 1 \\
x(1,7,8) & & + x(2,3,8) & & + x(3,7,8) & \geq & 1
\end{array}$$

and $k = 2$. Alternatively it can be obtained by the following argument: the set $\{3, 7, 8\}$ must be covered by the set $\{3, 7, 8\}$ or at least two sets among $\{1, 7, 8\}$, $\{2, 3, 4\}$, $\{2, 3, 8\}$ and $\{3, 4, 7\}$.

Such an interpretation holds for any inequality obtained with one iteration of Chvátal's rounding procedure with set $V' \subseteq V$ and number k:

$$\sum_{S \in \mathcal{S}_{max}} \alpha_S x_S \geq \left\lceil \frac{|V'|}{k} \right\rceil \tag{14}$$

with $\alpha_S = \left\lceil \frac{|V' \cap S|}{k} \right\rceil$, $\forall\ S \in \mathcal{S}_{max}$. In the following, such an inequality will be denoted by Chv(V',k). The next proposition gives a necessary condition for such an inequality to define a facet of $Conv(Cov(\mathcal{S}_{max}, V))$.

Proposition 5 *If Chv(V',k) defines a facet of $Conv(Cov(\mathcal{S}_{max}, V))$, then* $\left\lceil \frac{|V'|}{k} \right\rceil > \omega(G[V'])$.

Proof. Assume $\left\lceil \frac{|V'|}{k} \right\rceil \leq \omega(G[V'])$, let $C \subseteq V'$ be a maximum clique in $G[V']$, and let $v \in C$ and $w \in C$, $v \neq w$. Notice that v and w are adjacent since they both belong to the same clique C. Consider $\mathcal{S}_v = \{S \in \mathcal{S}_{max} | S \ni v\}$ and $\mathcal{S}_w = \{S \in \mathcal{S}_{max} | S \ni w\}$. A set that would be simultaneously in $\mathcal{S}_v$ and $\mathcal{S}_w$ would contain vertices v and w, which is impossible since it must be stable, hence $\mathcal{S}_v \cap \mathcal{S}_w = \emptyset$. This holds for any pair of vertices from C, and

$$\sum_{v \in C} \sum_{S \ni v} x_S = \sum_{S \cap C \neq \emptyset} x_S \geq |C| = \omega(G[V']) \tag{15}$$

is the sum of the cover inequalities of the vertices in c. Hence it is valid. But since all coefficients in it are 1, since $\alpha_S \geq 1\ \forall\ S$ such that $S \cap C \neq \emptyset$ and since $\omega(G[V']) \geq \left\lceil \frac{|V'|}{k} \right\rceil$, inequality (15) dominates inequality Chv(V',k). ■

In our example, this condition is fulfilled: $\left\lceil \frac{|\{3,7,8\}|}{2} \right\rceil = 2 > 1 = \omega(G[\{3, 7, 8\}])$.

5 Polyhedral results on the set packing formulation

We focus now on formulation $GC(Pack)$ and denote by $Pack(\mathcal{S}_2, V)$ its set of feasible solutions, i.e., $Pack(\mathcal{S}_2, V) = \{x \in \{0,1\}^{|\mathcal{S}_{max}|} : (9)\}$. Recall that we work with $\mathcal{S}_2$, the set of stable sets of size at least 2. With this formulation, all solutions consisting in only one stable set from $\mathcal{S}_2$ satisfy all constraints. There are as many such solutions as there are variables, and since they are affinely independent, $Conv(Pack(\mathcal{S}_2, V))$ is full-dimensional.

The *conflict graph* associated to the set $\mathcal{S}_2$ of stable sets is $\mathcal{G} = (\mathcal{S}_2, \{(S, S') : S \cap S' \neq \emptyset\})$. A clique in $\mathcal{G}$ is then a set of stable sets of G, having pairwise non empty intersections. We will say that a clique is maximal if it is maximal under inclusion, i.e., for a maximal clique $\mathcal{C} \subseteq \mathcal{S}_2$, and for each S in $\mathcal{S}_2 \backslash \mathcal{C}$ there is a $S' \in \mathcal{C}$ such that $S \cap S' = \emptyset$. The following result is proved in [Pad73] for general set packing polyhedra, and we adapt it here to $GC(Pack)$.

Proposition 6 *[Pad73] An inequality of the form*

$$\sum_{S \in \mathcal{C}} x_S \leq 1$$

defines a facet of $Conv(Pack(\mathcal{S}_2, V))$ if and only if $\mathcal{C}$ is a maximal clique in $\mathcal{G}$.

In the next section, we define and characterize a class of such inequalities which correspond to maximal cliques in $\mathcal{G}$.

5.1 Majority set cliques

Given a set of vertices $X \subseteq V$, we call *majority set* clique the subset $\mathcal{C}_X = \{S \in \mathcal{S}_2 : |S \cap X| \geq \frac{|X|+1}{2}\}$. Obviously, $\mathcal{C}_X$ is a clique. In what follows, we characterize which of them with $|X| = 1$ are maximal, then those with $|X| = 2$ and finally those with $|X| \geq 3$.

We call *single vertex cliques* the majority set cliques $\mathcal{C}_X$ with $|X| = 1$. Let $v \in V$. We assume that $|AN(v)| > 0$, since otherwise v is contained in no stable set of $\mathcal{S}_2$ and the single vertex clique $\mathcal{C}_{\{v\}}$ is empty.

Proposition 7 *Assume $AN(v) = \{w\}$. Then $\mathcal{C}_{\{v\}}$ is a maximal clique of $\mathcal{G}$ (in fact the isolated vertex $\{v, w\}$ of $\mathcal{G}$) if and only if $N(w) = N(v) = V \backslash \{v, w\}$.*

Proof. Notice first that $\{v, w\} \in \mathcal{C}_{\{v\}}$ and $N(w) \subseteq N(v)$, otherwise $|AN(v)| > 1$. If $N(w) = N(v)$, then any set $B \subseteq V$, with $B \neq \{v, w\}$ and $|B| \geq 2$ intersecting $\{v, w\}$ must contain at least one edge of G and hence $B \notin \mathcal{S}_2$, so $\mathcal{C}_{\{v\}}$ is maximal. If $N(w) \subset N(v)$, then $\{x, w\} \in \mathcal{S}_2$ for any $x \in AN(w) - \{v\}$ and $\mathcal{C}_{\{v\}}$ is not maximal. ■

Proposition 8 *Assume $|AN(v)| \geq 2$. Then $\mathcal{C}_{\{v\}}$ is a maximal clique in $\mathcal{G}$ if and only if $AN(v)$ is not a stable set in G.*

Proof. If $AN(v)$ is a stable set in G, then $AN(v) \in \mathcal{S}_2$ and $AN(v) \cap S \neq \emptyset \ \forall\, S \in \mathcal{C}_{\{v\}}$, i.e., $\mathcal{C}_{\{v\}}$ is not maximal. If there is an edge (x, y) in the subgraph induced by $AN(v)$, then any stable set $S \notin \mathcal{C}_{\{v\}}$ contains at most one vertex in the set $\{x, y\}$, and hence cannot intersect both $\{v, x\} \in \mathcal{C}_{\{v\}}$ and $\{v, y\} \in \mathcal{C}_{\{v\}}$. Thus $\mathcal{C}_{\{v\}}$ is maximal. ■

In the case where $AN(v)$ is a stable set (possibly of size one), the unique maximal clique containing $\mathcal{C}_{\{v\}}$ is $\mathcal{C}_{\{v\}} \cup \{S \in \mathcal{S}_2 : AN(v) \subseteq S\}$.

As mentioned in last section, if a graph G contains two non-adjacent vertices x and y such that $N(x) \subseteq N(y)$ (y dominates x), x can be removed from G without changing $\chi(G)$. In particular, if all such vertices are removed, no anti-neighborhood of a vertex can be a stable set. Consequently, in the resulting graph, all inequalities (9) define facets.

Assume $X = \{v, w\}$, and that v and w are not adjacent, since otherwise $\mathcal{C}_X$ is empty. Since $\frac{|X|+1}{2} = 1.5$, we have $\mathcal{C}_X = \{S \in \mathcal{S}_2 : \{v, w\} \subseteq S\}$. This implies that $\mathcal{C}_X \subseteq \mathcal{C}_{\{v\}}$ and $\mathcal{C}_X \subseteq \mathcal{C}_{\{w\}}$. We have as a consequence of Proposition 7, that if $N(v) = N(w) = V \backslash \{v, w\}$, then $\mathcal{C}_X = \mathcal{C}_{\{v\}} = \mathcal{C}_{\{w\}}$ is a maximal single vertex clique. This is the only case where $\mathcal{C}_X$ is maximal. Indeed, if at least one vertex in X, say v, has another non-neighbor, say u, then $\{u, v\} \in \mathcal{C}_{\{v\}}$, and $\mathcal{C}_X \subset \mathcal{C}_{\{v\}}$.

Assume now that $|X| \geq 3$. The next proposition gives a necessary and sufficient condition for $\mathcal{C}_X$ to define a maximal clique in $\mathcal{G}$.

Proposition 9 *Let $X \in V$ be a set of at least 3 vertices. Then the majority set clique $\mathcal{C}_X$ of $\mathcal{G}$ is maximal if and only if*

1. *$|X|$ is odd and*

2. *X is stable.*

Proof. If X satisfies 1 and 2, then for any $S \in \mathcal{S}_2 - \mathcal{C}_X$ we have $|X \cap S| \leq \frac{|X|-1}{2}$. Hence $|X \backslash S| \geq \frac{|X|+1}{2} \geq 2$ which means that $X \backslash S \in \mathcal{C}_X$, while $S \cap (X \backslash S) = \emptyset$. Thus $\mathcal{C}_X$ is maximal.

If X does not contain a stable set of cardinality larger than or equal to $\frac{|X|+1}{2}$ then $\mathcal{C}_X$ is empty (and hence not maximal).
Otherwise, let S be any stable set of $\mathcal{C}_X$. If X is not stable, let u and v be two vertices in X such that $(u, v) \in E$. Let $X' = X \backslash \{u, v\}$. Since S is stable, it contains either u or v, or none of them. Hence,

$$|S \cap X'| \geq |S \cap X| - 1 \geq \frac{|X| + 1}{2} - 1 = \frac{|X| - 1}{2} = \frac{|X'| + 1}{2}$$

which means that $S \in \mathcal{C}_{X'}$. Consequently, $\mathcal{C}_X \subseteq \mathcal{C}_{X'}$. Further, let $S' \in \mathcal{C}_X$ s.t. $|S'| = \frac{|X|+1}{2}$. Removing any vertex w from S' yields a stable set belonging to $\mathcal{C}_{X'}$, but not to $\mathcal{C}_X$, hence $\mathcal{C}_X \subset \mathcal{C}_{X'}$ and $\mathcal{C}_X$ is not maximal.
So X is a stable set and assume it has even cardinality. Let v be any vertex of X and S be a stable set of $\mathcal{C}_X$. Hence,

$$|S \cap (X \backslash \{v\})| \geq |S \cap X| - 1 \geq \frac{|X|+1}{2} - 1 = \frac{|X|-1}{2} = \frac{|X \backslash \{v\}|}{2}$$

which is not integer. So we also have $|S \cap (X \backslash \{v\})| \geq \frac{|X \backslash \{v\}|+1}{2}$, which means that $S \in \mathcal{C}_{X \backslash \{v\}}$ and $\mathcal{C}_X \subseteq \mathcal{C}_{X \backslash \{v\}}$. Further, let S' be a stable set of $\mathcal{C}_X$ with cardinality $\frac{|X|}{2} + 1$ such that $v \notin S'$ and $w \in S'$. Then $S' \backslash \{w\}$ belongs to $\mathcal{C}_{X \backslash \{v\}}$ but not to $\mathcal{C}_X$ which implies that $\mathcal{C}_X \subset \mathcal{C}_{X \backslash \{v\}}$ and $\mathcal{C}_X$ is not maximal. ∎

It follows that if one wants to find all majority set cliques which are not in the initial formulation, one needs only to consider the stable sets S with $|S| \geq 3$ and odd.

5.2 Other maximal cliques

There are lots of other maximal cliques in $\mathcal{G}$. Here are some illustrating examples.

Proposition 10 *Let X be a stable set of odd size at least 5, and $S \subset X$ such that $|S| = \frac{|X|+1}{2}$, $\mathcal{A} = \{T \in \mathcal{S}_2 : T \cap X = S\}$ and $\mathcal{B} = \{T \in \mathcal{S}_2 : T \cap X = X \backslash S\}$. Then $(\mathcal{C}_X \backslash \mathcal{A}) \cup \mathcal{B}$ is a maximal clique in $\mathcal{G}$.*

Proof. Since $\mathcal{C}_X$ is a clique, so is $\mathcal{C}_X \backslash \mathcal{A}$. Further we have clearly that each stable set in $\mathcal{C}_X \backslash \mathcal{A}$ intersects $X \backslash S$. So $(\mathcal{C}_X \backslash \mathcal{A}) \cup \mathcal{B}$ is a clique. To prove that it is maximal, consider a stable set $S' \notin (\mathcal{C}_X \backslash \mathcal{A}) \cup \mathcal{B}$. If $S' \in \mathcal{A}$, then obviously $S' \cap (X \backslash S) = \emptyset$, while $(X \backslash S) \in \mathcal{B}$ (notice that $|X \backslash S| \geq 2$). Assume now that $S' \notin (\mathcal{C}_X \cup \mathcal{B})$, and consider the stable set $(X \backslash S')$, which has empty intersection with S'. Since $|S' \cap X| \leq \frac{|X|-1}{2}$, $|X \backslash S'| \geq \frac{|X|+1}{2}$ which means that $(X \backslash S') \in \mathcal{C}_X$. Further, since $S' \notin \mathcal{B}$, $(X \backslash S') \notin \mathcal{A}$. So $(X \backslash S') \in \mathcal{C}_X \backslash \mathcal{A}$, which permits to conclude. ∎

Starting from a clique as defined in Proposition 10, one can do the same replacement with another set S' of size $\frac{|X|+1}{2}$ instead of S , provided that $(X - S') \cap (X - S) \neq \emptyset$. For doing the same a third time with S'' such that $|S''| = \frac{|X|+1}{2}$, one should ensure that $(X - S'') \cap (X - S) \neq \emptyset$ and that $(X - S'') \cap (X - S') \neq \emptyset$, and so on. Noticing that there are many possible choices for S, then for S' and so on, can give an idea of the huge number of maximal cliques in $\mathcal{G}$. Furthermore, there are other cliques which are neither majority set cliques, nor obtainable with the above construction. For instance in the graph displayed in Figure 3, the set

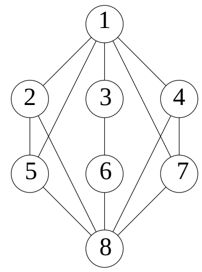

Figure 3: A graph with a clique facet which is not derived from a majority set clique.

$\mathcal{C} = \{\{2,3,4\},\{2,6,7\},\{3,5,7\},\{4,5,6\}\}$ is a maximal clique of $\mathcal{G}$ which does not belong to any previous case. The corresponding facet defining inequality is:

$$x(2,3,4) + x(2,6,7) + x(3,5,7) + x(4,5,6) \leq 1$$

Notice that this graph has no dominated vertex, so all single vertex cliques induce facets.

6 Exact graph coloring algorithms

The lower bound $\lceil \chi_f(G) \rceil$ on $\chi(G)$ can be used in a branch-and-bound algorithm for graph coloring. This is done in [MT96] with formulation $GC(Cov)$, where computational results show that the algorithm obtained is about the best actual exact coloring algorithm. The branching rule they choose is such that at each node of the enumeration tree, two subproblems are created, and they are of the same kind as the original one, i.e., graph coloring problems. Consequently, the algorithm can be called recursively. At the root node, an upper bound on $\chi(G)$ is obtained by a coloring obtained heuristically. We used a neighborhood search heuristic, with penalty evaporation to avoid cycling [Blö01]. This upper bound is updated each time a better coloring is found. For computing the lower bound, since the number of variables is very large, the use of column generation is unavoidable. The pricing problem is in this case a maximum weight stable set problem, where the graph is the instance graph, and the weight of each vertex is given by the dual value of the corresponding covering constraint at the optimal solution of the restricted linear program. Following Mehrotra and Trick's paper, we implemented a greedy heuristic which tries several times to find an improving column (or stable set of weight larger than 1). If at least one improving column is found, it is added to the restricted linear program which is optimized again with the simplex algorithm. If not, an exact algorithm is run, which will either find an improving column, or prove that no such column exists, in which case an optimal solution of the whole linear program has been reached.

The polyhedral results presented above could not directly be used to improve our algorithms. The characterization of facets with right hand side equal to 1 (Proposition 3) shows that $GC(Cov)$ is a good formulation since all its constraints induce facets. However, this result does not allow to derive new cutting planes. For Proposition 4, although the property of having a connected complement can be easily checked, we do not know if the instance graph G is $\chi-$critical and computing the corresponding inequality involves the knowledge of $\chi(G)$, so this result is of no practical use here. Concerning formulation $GC(Pack)$, we tried to add some majority set clique facets to the linear relaxations for small graphs, but the results were somewhat disappointing. We thus decided not to implement separation procedures for this kind of facets in the branch-and-price algorithm. We also inserted a procedure permitting to detect violated inequalities corresponding to odd holes in the conflict graph of stable sets. This can be done in polynomial time by solving $|V|$ shortest path problems, as is described in [GLS93]. Computational experiments showed that computation time tended to increase, while only a slight reduction in the number of visited nodes was sometimes observed. We did not use lifting procedures to obtain facets from the odd hole inequalities, and we are not able to say if it would provide better results.

Another class of valid inequalities express the fact that the number of stable sets intersecting a node in a subset V' of V must be at least $\chi(G[V'])$. Of course, if $\chi(G[V']) = \omega(G[V'])$, this inequality will never violate a solution of $GC^l(Cov)$. So we tried to systematically add such violated inequalities for subsets V', where $G[V']$ is a hole on five vertices. This did neither give us promising results, as we could obtain only small (< 0.1) improvements in the optimal solution value, even after adding several thousands of such inequalities, which substantially slowed down the LP optimization. The most persistent difficulty we encountered when adding cutting planes can be referred to as the problem symmetry: many times, when an inequality violating the current fractional solution is added, there is another such solution satisfying it, with the same value. Figure 4 displays a small graph G where such a situation happens. All maximal stable sets of this graph are of the form $\{i, i \bmod 5+1, i+5\}$, $\{i, i \bmod 5+1, i \bmod 5+6\}$ or $\{i, i \bmod 5+6, (i+3) \bmod 5+6\}$, for $i = 1\ldots,5$. An optimal solution, with value $\frac{10}{3}$, to $GC^l(Cov)$ is given by

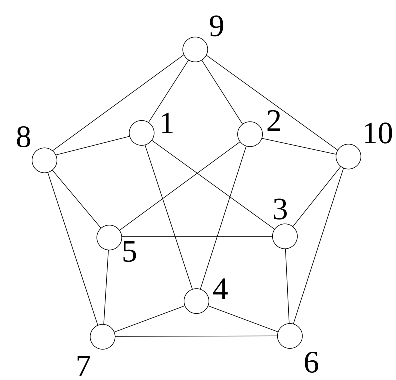

Figure 4: Graph G.

$$x(2,6,8) = \frac{2}{3}, \text{ and}$$

$$x(1,5,6) = x(1,5,10) = x(1,7,10) = x(2,3,7) = x(3,4,9)$$
$$= x(3,7,9) = x(4,5,9) = x(4,8,10) = \frac{1}{3}.$$

This solution does not satisfy the ineqality requiring that at least 3 stable sets have to cover the hole induced by the 5 vertices $\{1,3,4,5,7\}$. However, the symmetrical solution

$$x(3,7,9) = \frac{2}{3}, \text{ and}$$
$$x(1,2,7) = x(1,2,6) = x(1,5,10) = x(2,6,8) = x(3,4,8)$$
$$= x(4,5,10) = x(4,8,10) = x(5,6,9) = \frac{1}{3}$$

satisfies it, but violates the inequality corresponding to vertices $\{1,2,4,5,8\}$. Of course, adding these inequalities is not necessary in this example, since we still have $\lceil \chi_f(G) \rceil = \chi(G)$, but on larger graphs with larger gaps, this situation occurs even more frequently. Such symmetry problems are also encountered in [BHV00] for the origin-destination multicommodity flow problem, where the authors manage to efficiently combat it by adding some knapsack inequalities to the formulation. For our problem, it appears more difficult to find good classes of such symmetry-breaking inequalities. By computer-aided enumeration of the facets for small graphs, we have obtained the complete description of the corresponding polytopes $Conv(GC(Cov))$ and $Conv(GC(Pack))$. Unfortunately, the largest (interesting) graph for which $Conv(GC(Cov))$ (which has lower dimension than $Conv(GC(Pack))$) could be computed in reasonable time was `Myciel_3` (see Section 6.1.4), and has only 11 vertices. For this graph, the dimension of $Conv(GC(Cov))$ is 16, since `Myciel_3` has 16 maximal stable sets, and there are 226 facets. Furthermore, since inequalities describing holes on 5 vertices rarley appeared on those polytope descriptions, they are far from giving a good description of $Conv(GC(Cov))$. In [NP91], the authors get the same conclusions after testing a similar approach for edge coloring, which is equivalent to graph coloring restricted to line graphs. They further restrict attention to 3-regular graphs, and observe that edge inequalities and odd cycle inequalities (a cycle is a hole of the same size in the line graph) together are still not close to describing the convex hull of integer solutions.

We nevertheless obtained interesting results, as presented in the following. First, we show by numerical experiments that using formulation $GC(Pack)$ for computing $\chi_f(G)$ performs essentially as well as using $GC(Cov)$. Then we propose a simple preprocessing consisting in deleting vertices which are easily proven to be redundant, and show that it may somewhat improve the algorithm performance, if applied at each node of the branch-and-bound tree. Finally, we provide a cutting plane generation procedure for formulation $GC(Cov)$ and show that it often permits to make the enumeration tree substantially smaller, with limited loss in overall computation time.

The tests have been run on machines with a processor of 2 GHz, and the sizes of graphs have been chosen so that the algorithm finishes most times within an hour (the time limit). Linear programs are solved using the CPLEX 9.0 callable library from a program written in C++. All times are given in seconds. We would like to point out that taking into account the power ratio of our computers and those used in [MT96], it seems (although it is difficult to evaluate exactly) that our implementation is somewhat slower. This may partially be due to the way of solving the pricing problem exactly; we use here a basic branch-and-bound algorithm, and more effort has been invested for this sake in [MT96]. Anyway, the results we show next aim to give comparisons between variants of an algorithm, whose common parts are implemented in the same way.

6.1 Instance description

Let us first present the set of instance graphs which we used for running our tests.

6.1.1 Random graphs

The graphs of type `rand_n_p` are randomly generated on n vertices, such that for each pair of vertices, there is an edge connecting them with probability p. If p is close to 0 or 1, the lower bound $\lceil \chi_{frac(G)} \rceil$ is very often equal to $\chi(G)$, and its quality decreases as p approaches $\frac{1}{2}$. This makes `rand_n_p` with $p \simeq \frac{1}{2}$ the most difficult type of random graphs to color. For this reason, our tests have only been run with values $p = 0.3, 0.5\ 0.7$, and $n = 70, 80, 85, 90$, which is about the limit size beyond which actual exact algorithms do not finish in reasonable time.

6.1.2 Geometric graphs and reverse geometric graphs

A geometric graph of type `G_n_d` is constructed by uniformly generating n points in a square of side 1, and linking two vertices if their points are distant at most by d. The reverse geometric graph `RG_n_d` is obtained the same way, but by linking two vertices if their distance is *at least* d. Notice that if the set of points and a distance d are given, the corresponding geometric and reverse geometric graphs are complements of each other.We tested our algorithm first on geometric graphs with 500 vertices, $d = 0.1, 0.5, 0.9$, then on reverse geometric graphs with the same sizes and distances.

6.1.3 Queen graphs

The queen graph `queen_n_m` is obtained by associating a vertex to each square of an $n \times m$ chessboard, and linking two vertices a and b if a queen could move in one step from the square of vertex a to the square of vertex b. It is clear that

$\chi(\texttt{queen\_n\_m}) \geq \max(n, m)$, since the vertex set corresponding to a row or column is a clique of size n or m, respectively. It is known that if $n = m$ is not a multiple of 2 or 3, then $\chi(\texttt{queen\_n\_n}) = \max(n, m)$. Recently, the converse has been shown not to be true: for each $12 \leq n \leq 24$, $\chi(\texttt{queen\_n\_n}) = n$, see [Vas04]. So the most interesting queen_n_n graphs for our algorithm will be with $n \leq 10$, being a multiple of 2 or 3.

6.1.4 Mycielski graphs

In [Myc55], the following graph transformation is proposed. Given a graph $G = (\{x_1, \dots, x_n\}, E)$, construct a new graph $M(G)$ with vertex set

$$\{y_1, \dots, y_n, z_1, \dots, z_n, w\}$$

and edge set such that $\{z_1, \dots, z_n\}$ is a stable set, y_i is linked to y_j if and only if x_i is linked to x_j, y_i is linked to z_j if and only if x_i is linked to x_j and w is linked to all z_i. It is not difficult to prove that $\chi(M(G)) = \chi(G) + 1$, while $\omega(M(G)) = \omega(G)$. Hence this transformation permits to obtain graphs with arbitrarily large gaps between chromatic and clique numbers. The graph Myciel_1 is K_2, so Myciel_2$=M(K_2) = C_5$, Myciel_3 is a graph with chromatic number 4 and clique number 2, and more generally $\chi(\texttt{Myciel\_k}) = k + 1$.
It is shown in [LPU95] that

$$\chi_f(\texttt{Myciel\_k}) = \chi_f(\texttt{Myciel\_k-1}) + \frac{1}{\chi_f(\texttt{Myciel\_k-1})}.$$

So for the smaller Mycielski graphs:

$$\chi_f(\texttt{Myciel\_2}) = 2.5, \chi_f(\texttt{Myciel\_3}) = 2.9,$$

$$\chi_f(\texttt{Myciel\_4}) \simeq 3.24, \chi_f(\texttt{Myciel\_5}) \simeq 3.55$$

and so on. The gap $\chi(\texttt{Myciel\_k}) - \lceil \chi_f(\texttt{Myciel\_k}) \rceil$ becomes thus arbitrarily large as k increases. This makes Mycielski graphs the most difficult graphs (with a given number of vertices) to color of our whole instance set.

6.2 Comparisons of using $GC(Cov)$ and $GC(Pack)$

The variant presented in this section is about the same algorithm as in [MT96], except the way of computing $\chi_f(G)$, since the formulation used is $GC(Pack)$. As mentioned in Section 2, the set of variables is $\{x_S : s \in \mathcal{S}_2\}$ and is much larger than $\mathcal{S}_{max}$. However, since the number of constraints is also $|V|$, no more than $|V|$

variables will have a strictly positive value at each basic solution, and the computational time required to solve the linear program to optimality is comparable to the time required with $GC(Cov)$. Comparisons of execution time and size of enumeration tree between both branch-and-price algorithms are given in Table 1. All times are given in seconds, and the columns "Nodes" contain the number of nodes in the enumeration tree. For random, geometric and reverse geometric graphs, all values are averages over 10 graphs, and in brackets are the number of instances solved within one hour, if less than 10. Only one Mycielski graph appears in the tables, since no version of our algorithm could solve `Myciel_5` (which has only 47 vertices). This is clearly due do the large gap between $\chi(G)$ and $\chi_f(G)$, leading to a large enumeration tree.

			$GC(Cov)$		$GC(Pack)$	
Graph	χ	χ_f	Time	Nodes	Time	Nodes
rand_70_0.3	7.88	6.83	489(9)	2022	301(8)	805
rand_70_0.5	11.8	10.7	41.3	402	37.3	278
rand_70_0.7	17.1	16.3	12.9	7.4	2.2	4.4
rand_80_0.3	8.1	7.35	225	443	226	351
rand_80_0.5	12.7	11.6	178(9)	1197	240(9)	1083
rand_80_0.7	19	17.9	23.9	151	34	402
rand_85_0.3	8.66	7.66	22.5(2)	1	14.3(2)	1
rand_85_0.5	13	12.0	57.2(9)	204	46(9)	143
rand_85_0.7	19.8	18.7	37.1	315	29.4	237
rand_90_0.3	9	7.94	239(6)	130	215(6)	127
rand_90_0.5	13.8	12.5	160(5)	642	722(6)	2177
rand_90_0.7	20.5	19.35	78.3	801	64	459
g_300_0.1	10.4	10.4	11	1	37.3	1
g_300_0.5	80.5	80.3	348	121	444	106
g_300_0.9	206	206	487	1	114	1
gr_300_0.1	71.7	71.3	76.7	5.8	47.3(8)	22.7
gr_300_0.5	7.1	6.82	56	70	1670(5)	1
gr_300_0.9	3.8	3.67	31.1	1	1390	1
queen8_8	9	8.44	7.45	1	1.67	1
queen8_9	9	9	–	–	1520	4677
queen9_9	10	9	78.7	55	34.3	21
myciel_4	5	3.24	1.35	659	3.36	993

Table 1: Performances with $GC(Cov)$ and $GC(Pack)$.

Results are mixed. While the version with $GC(Pack)$ seems to work slightly better on queen graphs, it is much less efficient on reverse geometric graphs and results are comparable on geometric and random graphs. More generally, it seems that $GC(Pack)$ is more appropriate for graphs with high density; for graphs with low density and large stable sets, $GC(Cov)$ becomes more efficient. Notice that both algorithms differ only by the way of computing $\chi_f(G)$. Those results can hence be explained by the fact that $GC(Pack)$ involves as many variable as there are stable sets in G, while $GC(Cov)$ involves a variable per *maximal* stable set. So the additional number of variables in $GC(Pack)$ becomes much larger for these graphs with low density and large stable sets. Table 2 gives the approximate values of $\alpha(G)$ for our instance set. Those results permit us anyway to assert that $GC(Pack)$ is not worse than $GC(Cov)$ for using in a branch-and-price approach. Since there are more results in the literature about set packing polytopes, as compared to the set covering polytopes, many extensions to the presented algorithm may be tried.

Type of G	$\sim \alpha(G)$
`rand_n_0.3`	14
`rand_n_0.5`	9
`rand_n_0.7`	6
`g_300_0.1`	71
`g_300_0.5`	6
`g_300_0.9`	3
`gr_300_0.1`	10
`gr_300_0.5`	80
`gr_300_0.9`	206
`queen_n_m`	min(n,m)
`myciel_4`	11

Table 2: Approximate values of $\alpha(G)$ for our instance set.

6.3 Preprocessing

Here we see how one can sometimes slightly speed up the algorithm by applying two simple vertex deletion rules at each node of the branch-and-bound tree. The first rule is built on the notion of domination defined in Section 4. Recall that a vertex v dominates a vertex w if $N(w) \subset N(v)$, and a dominated vertex can always be removed from the graph without decreasing the chromatic number. The second rule requires the knowledge of a lower bound $\underline{\chi}$ on $\chi(G)$. A vertex v such that $d(v) < \underline{\chi} - 1$ can also be removed from the graph, since there is always a color in the set $\{1, \ldots, \underline{\chi} - 1\}$, available for vertex v, after having optimally colored $G[V \setminus \{v\}]$. This last reduction is worth being applied, as we have a good lower bound on $\chi(G)$. Hence the following simple procedure sometimes permits one to reduce the graph G, without changing $\chi(G)$.

Input: Graph G, lower bound $\underline{\chi}$ on $\chi(G)$
Output: Possibly reduced graph G', s.t. $\chi(G') = \chi(G)$.

1: Set $G' = G$;
2: **repeat**
3: Remove each dominated vertex from G';
4: Remove each vertex v such that $d(v) < \underline{\chi} - 1$;
5: **until** no more vertex can been removed this way.

Since at each node of the branch-and-bound tree the problem is a graph coloring one, this preprocessing can be called each time a new subproblem has been created. The complexity of checking if a node is dominated being roughly in $O(|V|^2)$, each loop's complexity is in $O(|V|^3)$. This is small as compared to the time necessary to compute the lower bound $\chi_f(G)$, which requires to solve at least one maximum weight stable set problem on G to optimality.

The computation time and number of nodes in the enumeration tree with and without preprocessing are presented in Table 3. Formulation $GC(Cov)$ is used, and results without proprocessing are reported from Table 1. The additional column "Deleted" shows the average (over all nodes in the enumeration tree) number of deleted vertices due to the preprocessing.

			No preprocessing		With preprocessing		
Graph	χ	χ_f	Time	Nodes	Time	Nodes	Deleted
`rand_70_0.3`	7.88	6.83	489(9)	2022	481(9)	2070	0.0022
`rand_70_0.5`	11.8	10.7	41.3	402	37.1	395	0.0025
`rand_70_0.7`	17.1	16.3	12.9	7.4	12.8	6.8	0.0735
`rand_80_0.3`	8.1	7.35	225	443	209	419	0.0035
`rand_80_0.5`	12.7	11.6	178(9)	1197	170(9)	1190	0.0005
`rand_80_0.7`	19	17.9	23.9	151	22.2	146	0.0238
`rand_85_0.3`	8.66	7.66	22.5(2)	1	22.5(2)	1	0
`rand_85_0.5`	13	12.0	57.2(9)	204	55.5(9)	194	0.0005
`rand_85_0.7`	19.8	18.7	37.1	315	37.0	302	0.0353
`rand_90_0.3`	9	7.94	239(6)	130	238(6)	130	0
`rand_90_0.5`	13.8	12.5	160(5)	642	161(5)	642	0
`rand_90_0.7`	20.5	19.35	78.3	801	83.1	839	0.0219
`g_300_0.1`	10.4	10.4	11	1	11.4	1	10.4
`g_300_0.5`	80.5	80.3	348	121	280	63.2	0.86
`g_300_0.9`	206	206	487	1	348	1	17.2
`gr_300_0.1`	71.7	71.3	76.7	5.8	23.7	2.4	77
`gr_300_0.5`	7.1	6.82	56	70	1.9	1.2	222
`gr_300_0.9`	3.8	3.67	31.1	1	0.41	1	295
`queen8_8`	9	8.44	7.45	1	7.31	1	0
`queen8_9`	9	9	–	–	–	–	–
`queen9_9`	10	9	78.7	55	78.9	55	0
`myciel_4`	5	3.24	1.35	659	1.13	517	1.34

Table 3: Performances with and without preprocessing.

Results on random graphs indicate that it is worth running the preprocessing. Although there are only few graphs where vertices could be deleted, some time has been saved, and the enumeration tree tends to be somewhat smaller. Notice that for some instances the number of visited nodes has increased, when only a few vertices have been deleted. On geometric graphs, a substantial number of vertices are removed by the preprocessing, which is a direct consequence of the structure of the graph: vertices near a corner have a good chance of having a small degree, or of being dominated. For reverse geometric graphs the effect is even stronger: in most cases, more than half the graph is removed, and the larger the parameter `d`, the more vertices are removed, what is not difficult to understand. For queen graphs, results show that the preprocessing brings no improvement. This is not surprising, as vertices have all about the same degree and symmetrical characteristics. For Mycielski graphs, the preprocessing brings some improvement, although `Myciel_5` could still not be solved within one hour.

To summarize, the computation times are not systematically better, but they are worse in only few cases. Indeed, as compared to the column generation procedure, the preprocessing takes very little time. Since it relies on simple detection rules, it is not difficult to implement. So it is worth inserting a call to such a procedure at each node of the branch-and-bound tree of any graph coloring algorithm, provided that the subproblems are still graph coloring problems. Finally, we observed in our experiments, that nodes were deleted thanks to the domination criterion rather than to the low degree criterion. So even if no good lower bound is known, it may be applied successfully.

6.4 A cutting plane procedure for $GC(Cov)$

We present here a cutting plane generation procedure, which is the only one that gave us satisfying results. It consists in detecting some violated $(0, \frac{1}{2})$-inequalities for the $GC(Cov)$ formulation.

Specifically, a relaxation of $GC(Cov)$ is considered in which the original constraint

$$\sum_{S \in \mathcal{S}_{max} : v \in S} x_S \geq 1$$

corresponding to each vertex v is replaced by a set of constraints

$$x_{S_i} + x_{S_j} + 2 \sum_{S \in \mathcal{S}_{max} \setminus \{S_i, S_j\} : v \in S} x_S \geq 1 \; \forall \; S_i, S_j \in \mathcal{S}_{max} : v \in S_i \cap S_j$$

One can determine in polynomial time whether there exists a (0,1)-inequality based on this new system $GC_{rel}(C)$ which is violated by the current solution x^* since $GC_{rel}(C)$ contains at most two odd coefficients per row, see [CF96]. This separation problem amounts to finding a minimum-weight odd cycle C^* in the conflict graph $\mathcal{G} = \{\mathcal{S}_{max}, \{(S, S') : S \cap S' \neq \emptyset\}\}$, in which the weight of an edge (S_i, S_j) is equal to

$$x^*_{S_i} + x^*_{S_j} - 1 + 2 \min_{v \in S_i \cap S_j} \{ \sum_{S \in \mathcal{S}_{max} \setminus \{S_i, S_j\} : v \in S} x^*_S \}.$$

If the minimum weight odd cycle C^* has a weight smaller than one, the $(0, \frac{1}{2})$-Chvátal-Gomory cut (CG cut) involving the inequalities of $GC_{rel}(C)$ defining the edges of C^* is violated. The $(0, \frac{1}{2})$-CG cut involving the corresponding original inequalities, i.e., from $GC(Cov)$, is also violated by the current solution and we thus add it to the current program.
Note that only stable sets S s.t. $0 < x^*_S < 1$ need to be considered when solving the minimum weight odd cycle problem. Indeed, the weight of an edge linking x_{S_i} ans x_{S_j} is

$$x^*_{S_i} + x^*_{S_j} - 1 + 2 \min_{v \in S_i \cap S_j} \{ \sum_{S \in \mathcal{S}_{max} \setminus \{S_i, S_j\} : v \in S} x^*_S \} \geq |x^*_{S_i} + x^*_{S_j} - 1|.$$

Hence the total weight of any odd cycle C containing a vertex (stable set) S is larger than $|2x^*_S - 1|$, which implies that the corresponding $(0, \frac{1}{2})$-CG cut is not violated if $x^*_S = 0$ or 1.

Unfortunately, the pricing problem is not exactly a maximum weight stable set problem anymore. However, the new problem structure remains similar to the maximum stable set problem as will be shown now.

Each cutting plane corresponds to a subset of vertices of odd cardinality. Denote by $\mathcal{H} = \{H_1, \ldots, H_p\}$ the set of those odd-sized subsets. At the optimum of the restricted linear program, the reduced cost for a nonbasic variable (stable set) S is given by

$$\mathbf{1} - \sum_{i \in S} \lambda_i - \sum_{i=1}^{p} \mu_i \lceil \frac{|S \cap H_i|}{2} \rceil,$$

where $\lambda_1, \ldots, \lambda_{|V|}$ are the dual variables corresponding to the vertex cover constraints and $\mu_1, \ldots, \mu_p$ are the dual variables corresponding to the generated cutting planes. Here $\lceil \frac{|S \cap H_i|}{2} \rceil$ is the coefficient of the variable corresponding to S, in the constraint corresponding to H_i. Since $GC(Cov)$ is a minimization problem, a variable x_S may enter the basis if and only if its reduced cost is strictly lower than 0, which is achieved if and only if

$$\sum_{i \in S} \lambda_i + \sum_{i=1}^{p} \mu_i \lceil \frac{|S \cap H_i|}{2} \rceil > 1.$$

In other words, the pricing problem consists in finding a stable set S with objective larger than 1, where the objective is the sum over S of the weights, plus a "bonus" μ_i for each subset H_i. This bonus is counted once if S has 1 or 2 vertices in H_i, twice if S has 3 or 4 such vertices, three times if S has 5 or 6 such vertices, and so on. Although this is a nonlinear contribution to the objective, it is not difficult to take into account in the exact (branch-and-bound) algorithm, and has a good linear approximation that can be used in the heuristic procedures.

Applying this to our graphs, we observed that the lower bound on $\chi(G)$ was increased in most cases by less than 0.1. So we decided to run the cutting plane algorithm only on those nodes where increasing the lower bound by 0.1 could permit to backtrack, i.e., where $\chi_{sup}(G)-1-t \leq \chi_f(G') \leq \chi_{sup}(G)-1$, with $\chi_{sup}(G)$ being the value of the best coloring found so far, and t a threshold value we fixed at 0.1. Though the increase in the lower bound is small, we could obtain interesting results, thanks to the quickness of our plane procedure. In Table 4, column "Cuts" contains the average (over all nodes in the enumeration tree) number of cuts added to the formulation, and under "Back." are the numbers of nodes where the lower bound on $\chi(G)$ could be sufficiently increased to permit to backtrack.

			No Cuts		With cuts			
Graph	χ	χ_f	Time	Nodes	Time	Nodes	Cuts	Back.
rand_70_0.3	7.88	6.83	489(9)	2020	655(8)	821	2.11	29
rand_70_0.5	11.8	10.7	41.3	402	55.8	250	5.34	26.2
rand_70_0.7	17.1	16.3	12.9	7.4	13.1	3.4	5.47	0.4
rand_80_0.3	8.1	7.35	225	4438	20.1(9)	2.55	0.35	0.11
rand_80_0.5	12.7	11.6	178(9)	1197	258(9)	833	5.53	75.4
rand_80_0.7	19	17.9	23.9	151	27.2	68.2	9.17	9.6
rand_85_0.3	8.5	7.63	22.5(2)	1	22.4(2)	1	0	0
rand_85_0.5	13	12.0	57.2(9)	204	86.6(9)	139	6.03	11.1
rand_85_0.7	19.8	18.7	37.1	315	37.1	124	7.75	24.6
rand_90_0.3	9	7.94	239(6)	130	264(6)	125	0.57	1.33
rand_90_0.5	13.8	12.6	160(5)	642	204(5)	377	5.6	33.4
rand_90_0.7	20.5	19.3	78.3	801	85.3	313	9.25	60.3
g_300_0.1	10.4	10.4	11	1	11.3	1	0	0
g_300_0.5	80.5	80.3	348	121	380	123	2.41	0
g_300_0.9	206	206	487	1	487	1	0	0
gr_300_0.1	71.7	71.3	76.7	5.8	76.3	5.8	0.1	0.1
gr_300_0.5	7.1	6.82	56	70	53.3	30.6	0.63	2.2
gr_300_0.9	3.8	3.67	31.1	1	31.1	1	0	0
queen8_8	9	8.44	7.45	1	7.33	1	0	0
queen8_9	9	9	–	–	–	–	–	–
queen9_9	10	9	78.7	55	89.2	31	10.1	4
myciel_4	5	3.24	1.35	659	1.88	339	2.79	88

Table 4: Performances with and without adding cutting planes.

As can be seen, the average computation time is not better, but the enumeration tree is often much smaller. Those results are encouraging, since there are several points which may be improved in our algorithm. First, the cutting planes generated at a given node are not kept in the descendant nodes. Although the set of variables change, since the graph is modified, there should be a way of keeping at least some partial information on the cuts generated. Further, we did not try values different from 0.1 for the threshold t. In particular, if some information about cutting planes is kept when branching, a larger value for t may bring better results.

7 Conclusion

Two formulations of the graph coloring problem involving an exponential number of variables are explored. They are a covering formulation, already considered by [MT96] and a new packing formulation. Several families of facets are characterized: inequalities in the initial formulations and further inequalities derived from a lemma of [Sas89] and maximal cliques in the conflict graph of stable sets. Necessary or sufficient conditions for additional classes of facets are also given. Computational results with branch-and-cut-and-price algorithms show both formulations to be about equally efficient. Preprocessing based on vertex deletion proved to be useful when applied at each node of the branch-and-bound tree. A facet generating procedure for the set covering formulation, while not reducing computation time substantially, entailed a reduction in the size of the enumeration tree.

References

[BHV00] C. Barnhart, C. A. Hane, and P. H. Vance. Using branch-and-price-and-cut tu solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, Mar-Apr 2000.

[Blö01] I. Blöchliger. A new heuristic for the graph coloring problem. Master's thesis, École Polytechnique Fédérale de Lausanne, 2001.

[CF96] A. Caprara and M. Fischetti. $\{0,\frac{1}{2}\}$-Chvátal-Gomory cuts. *Mathematical Programming*, 74:221–235, 1996.

[Chv73] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:199–216, 1973.

[CMaZ02] P. Coll, J. Marenco, I. Méndez Díaz, and P. Zabala. Facets of the graph coloring polytope. *Annals of Operations Research*, 116(1-4):79–90, 2002.

[CS89] G. Cornuéjols and A. Sassano. On the 0,1 facets of the set covering polytope. *Mathematical Programming*, 43:45–55, 1989.

[GLS93] M. Grötschel, L. Lovàsz, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, 1993.

[LPU95] M. Larsen, J. Propp, and D. Ullman. The fractional chromatic number of Mycielski's graphs. *J. Graph Theory*, 19:411–416, 1995.

[MT96] A. Mehrotra and M. A. Trick. A column generation approach for graph coloring. *INFORMS, Journal on Computing*, 8(4):344–354, 1996.

[Myc55] J. Mycielski. Sur le coloriage des graphes. *Colloquium Mathematicum*, 3:161–162 (in French), 1955.

[NP91] G. L. Nemhauser and Sungsoo Park. A polyhedral approach to edge coloring. *Operations Research Letters*, 10:315–322, 1991.

[Pad73] M. W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

[Sas89] A. Sassano. On the facial structure of the set covering polytope. *Mathematical Programming*, 44:181–202, 1989.

[Sch97] E. R. Scheinerman. *Fractional Graph Theory*. Wiley-Interscience, 1997.

[Sch03] A. Schrijver. *Combinatorial Optimization*, volume B. Springer, 2003.

[Sch04] D. Schindl. *Some Combinatorial Optimization Problems in Graphs with Applications in Telecommunications and Tomography*. PhD thesis, École polytechnique fédérale de Lausanne, 2004.

[Vas04] M. Vasquez. New results on the Queens_n^2 graph coloring problem. *Journal of Heuristics*, 10(4):407–413, July 2004.