A New Decoposition Method for Chromatic Number Problem

Khellouf leila <khelloufleila@yahoo.fr>

April 11, 2017

Contents

Abstract

A new decomposition method that uses recursive steps is proposed to solve the vertex chromatic number problem. Some interessting properties with their proofs are put forward to show how a graph can be efficiently decomposable and its chromatic number determined or bounded. We propose an hybrid decomposition algorithm based on our method with Branch-Price and Cut method to give approximate values of chromatic number where has produced several good results for the well-known DIMACS banchmark graphs.
Key words: Graph Colouring, Graph Decomposition, Chromatic Number, Branch-Price and cut, Meta-Heuristics [1],

1 Introduction

Garey and al. in 1979 demonstrated that the k-coloring problem is NP-complete and that the determination of the chromatic number $\chi(G)$ is NP-hard(Garey and Johnson, 1979). Therefore several methods and heuristics have been proposed to solve this problem.
Mechanisms for the decomposition of graphs have been extensively studied by many researchers for the last three decades. These include the modular decomposition by Mehring and Radermacher(1984), the tree decomposition designed by Robertson and Seymour(1986), and the decomposition by clique separator of Tarjan(1985).
We use the graph colouring problem as a platform to test our decomposition methodology. this combinatorial optimisation problem is one of the most known challenging problems in the literature along side the travelling salesman problem. This problem has a wide range of applications including timetabling and scheduling.
A colouring of the vertices of a graph is a mapping that assigns one colour to each vertex in such a way that any two adjacent vertices have distinct colours. A colouring is optimal if it uses the least number of colours which is referred to as the chromatic number $\chi(G)$ of the graph G, Malaguti and Toth(2009).
Various decomposition methods have been proposed for this NP-hard problem. For instance, Lucet et al.(2006) propose a method based on linear decomposition for the graph whereas very recently Rao(2008) put forward an interesting scheme that uses a split decomposition to optimally colour a graph. Ekim and Gimbel(2009) examine some relations between cochromatic, split-chromatic and chromatic numbers using the partitions of the vertex set notion.
Our approach differs from theirs in the following: our idea is to use successive decomposition of the graph that will obviously produce successive sub-graphs or blocks for which their chromatic number is known. Such graphs include the bipartite graphs, the perfect K4-free graphs by Tucker(1987),the HHD-free

graphs by Jamison and Olariu(1988), the weakly triangulated graphs by Hayward et al.(1989), the Meyniel graphs by Hertz(1990), the p4âĹŠfree graphs by Babel(1997), among others. In the others hand, A.Haddadene et al. (1998) propose a sequential method with 3-chromatic interchange for colouring some perfect graphs. In other words, we follow this mechanism until we reach the final step of our decomposition and hence we recursively bachtrack to derive the optimal solution to the original graph.
The paper is organised as follows: the next section present our methodology followed by some new results and proofs. An application to special classes and for some instance DIMACS data sets is provided in section 4 for illustration purposes.section 5 presents a Branch and Price algorithm to the Minimum Graph Coloring Problem and reports additional computational results. Finally in the last section Conclusios and suggestions are given.

2 Methodology

Indeed, each graph can be decomposable into two sub-graphs say G_1 and $G_1^{'}$ whare the chromatic number of G_1 can be known whereas the chromatic number of $G_1^{'}$, is unkown. We perform such a procedure of decomposition(or splitting) for the resulting graph $G_1^{'}$. This proces is then repeated until the sub-graph $G_{k-1}^{'}$ that can be decomposable into two sub-graphs G_k and $G_k^{'}$ with known chromatic number. Figure 1 is given to illustrate this recursive process. The graph G can be decomposable into a set of 2k-subgraphs $L = \{G_1, G_1^{'}, G_2, G_2^{'}, ..., G_k, G_k^{'}/\chi(G_k^{'})\}$ is known and $\chi(G_i)$ are unkown,$\forall i \in 1, 2, ..., k$. The idea is if we can obtain the chromatic number of the two blocks G_k and $G_k^{'}$, we can then obtain the chromatic number of $G_{k-1}^{'}$. Using the same token we can therefore recursively find chromatic number for the original graph G. In this case, the chromatic number problem reduces to the chromatic number of the graph which is decomposable into two blocks with their known respective chromatic numbers. The aim of the paper is therefore to get some theoretical results for the chromatic number of a class of graphs that can be decomposable into two blocks for their individual chromatic number exists.

3 Properties

Let $G = (V, E)$ be a graph which is decomposable into two blocks G_1 and G_2 for which both chromatic numbers $\chi(G_1)$ and $\chi(G_2)$ are known.
As $\max(\chi(G_1), \chi(G_2)) \leq \chi(G)$, this property can provide us with a lower bound for $\chi(G)$.
Let:
$F_1 = \{v \in V(G_1)/\exists v^{'} \in V(G_2) : vv^{'} \in E(G)\}$
$F_2 = \{v \in V(G_2)/\exists v^{'} \in V(G_1) : vv^{'} \in E(G)\}$
$F = G(F_1 \cup F_2)$ be a subgraph induced by the vertices in $(F_1 \cup F_2)$ and $\alpha(\overline{F})$

be the cardinal of the maximum stable set in $\overline{F}$, $\mid V(G) \mid$, denote the order of graph G.
In the following we present some properties for which the obtained sub-graphs have known characteristics.

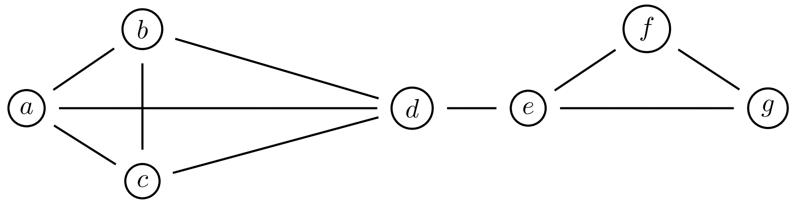
Property 3.1. *if F_1 and F_2 are cliques then $\chi(G) = \max(\chi_1, \chi_2, \alpha(\overline{F}))$*

Proof. F_1 is an articulation clique in G because F_1 is a clique and $GG(F_1)$ is a graph that can be decomposable into two connected components. Similarly, F_2 is also an articulation clique in G_2 as F_2 is a clique and $GG(F_2)$ is a graph that can be decomposable into two connected components.
Due to the theorem of Tarjan (1985) we have the following:
$\chi(G) = \max(\chi_1, \chi(G_2^{'}))$ avec $\chi(G_2^{'}) = G_2 \cup G(F_1)$.
$\chi(G_2^{'}) = \max(\chi_2, \chi(G(F_1 \cup F_2))) = \max(\chi_1, \chi(F))$.

Give that F_1 is a clique, hence $\overline{F_1}$ is a stable set. Similarly F_2 is a clique so $\overline{F_2}$ is a stable set also.
Note that $G(\overline{F_1 \cup F_2}) = \overline{F}$ is bibartite graph which is also a perfect graph, (see Ford et Fulkerson. (1962)) so $\chi(F) = \theta(\overline{F}) = \alpha(\overline{F})$
Thus, $\chi(G) = \max(\chi_1, \max(\chi_2, \alpha(\overline{F}))) = \max(\chi_1, \chi_2, \alpha(\overline{F}))$ □

An illustrative example

Figure 2 shows a graph which is decomposable into two G_1 and G_2 where $\chi(G_1) = 3$ and $\chi(G_2) = 3$ such that F_1 and F_2 are cliques. Using property 1, we get $\chi(G) = 4$ as shown in Figure 2.



Corollary 3.1. *Let G be a graph decomposable into two blocks G_1 and G_2 with their chromatic number $\chi(G_1)$ and $\chi(G_2)$, respectively.*
If $V(G) = \{u_1, u_2, ..., u_{|V(G_1)|}\}$ is a stables set and $\mathcal{N}(u_i)$ is a clique, $\forall i \in \{1, 2, ..., \mid V(G_1) \mid\}$, then $\chi(G) = \max(\chi_2, \mid \mathcal{N}(u_1) \mid +1, \mid \mathcal{N}(u_2) \mid +1, ..., \mid \mathcal{N}(u_{|V(G_1)|}) \mid +1)$

Proof. Let $V(G) = \{u_1, u_2, ..., u_{|V(G_1)|}\}$ be a set of vertices of graph G_1. To determine $\chi(G)$, we use the following decomposition as shown in Figure 3.

$\mathcal{N}(u_{|V(G_1)|})$ is a clique in a graph G. But $V(G_1)$ is a stable set so $\mathcal{N}(u_{|V(G_1)|})$ is a clique in the graph G_2. Using the property 1 we find that
$\chi(G_2 \cup G_1(\{u_{|V(G_1)|}\})) = \max(\chi(G(\{u_{|V_G|}\})), \chi_2, \alpha(\overline{F}_{|V(G)|})) = \max(1, \chi_2, \alpha(\overline{F}_{|V(G_1)|}))$
where $F = G(\{u_{|V(G_1)|}\} \cup \mathcal{N}(u_{V(G_1)}))$
since $\alpha(\overline{F}_{|V(G_1)|}) = \mid \mathcal{N}(u_{V_{G_1}}) \mid +1$,

$F = G(\{u_{|V(G_1)|}\}) \cup \mathcal{N}(u_{|V(G_1)|})$ is a clique so $\chi(G_2 \cup G_1(\{u_{|V_{G_1}|}\})) = \max(\chi_2, |\mathcal{N}(u_{|V(G_1)|}) + 1)$
Recursively for every $i \in \{1, ..., | V(G_1) |\}$, $\mathcal{N}(u_i)$ is a clique in the graph G but $V(G_1)$ is a stable set so $\mathcal{N}(u_i)$ is a clique in the graph
$G_2 \cup G_1(\{u_{i+1}, u_{i+2}, ..., u_{|V(G_1)|}\})$. Using the property 1 we find that
$\chi(G_2 \cup G_1(\{u_i, u_{i+1}, ..., u_{|V(G_1)|}\})) = \max(1, \chi_2, \alpha(\overline{F}_i), ..., \alpha(\overline{F}_{|V(G_1)|}))$ ou $\forall k \in \{i, ..., | V(G_1) |\} : F = G(\{u_k\} \cup \mathcal{N}(u_k))$ or $\alpha(\overline{F}_k) =| \mathcal{N}(u_k) | +1$
$F_k = G(\{u_k\} \cup \mathcal{N}(u_k))$ is a clique so
$\chi(G_2 \cup G_1(\{u_i, u_{i+1}, ..., u_{|V(G_1)|}\})) = \max(\chi_2, | \mathcal{N}(u_i) | +1, | \mathcal{N}(u_{i+1}) | +1, ..., | \mathcal{N}(u_{V_{G_1}}) | +1)$
For $i = 1$ we have
$\chi(G_2 \cup G_1(\{u_1, u_2, ..., u_{|V(G_1)|}\})) = \max(\chi_2, | \mathcal{N}(u_1) | +1, | \mathcal{N}(u_2) | +1, ..., | \mathcal{N}(u_{|V_{G_1}|}) | +1)$ so
$\chi(G) = \max(\chi_2, | \mathcal{N}(u_1) | +1, | \mathcal{N}(u_2) | +1, ..., | \mathcal{N}(u_{|V_{G_1}|}) | +1)$.

□

Property 3.2. *If F_1 is a stable set and F_2 is not, then*
$\max(\chi_1, \chi_2) \leqslant \chi(G) \leqslant \max(\chi_1, \chi_2) + 1$

Proof. Since F_1 is a stable set then $\forall(v, v') \in F_1^2$, $vv' \notin E(G)$
Let G^c be a graph obtained after the contraction of all vertices of F_1 on vertex v_0. As v_0 is an articulation vertex in the graph G^c, then:

$$\chi(G^c) = \max\lfloor \chi(G_1^c), \chi((G_2 \cup G(F_1))^c) \rfloor \tag{1}$$

with G_1^c (resp $(G_2 \cup G(F_1))^c$) being a graph that can be obtained after the contraction of all vertices of F_1 in the graph G_1 (resp: in the graph $G_2 \cup G(F_1)$) we therefor have

$$\chi(G_1) = \chi_1 \leqslant \chi(G_1^c) \tag{2}$$

As G_2 is a subgraph of $(G_2 \cup G(F_1))^c$, then

$$\chi(G_2) = \chi_2 \leqslant \chi((G_2 \cup G(F_1))^c) \tag{3}$$

However the graph G_1^c can be coloured using $\chi(G)$ colours. this is carried out by assigning to each vertex $v \in G_1^c - \{v_0\}$ its initial colour in $\chi(G_1)$ coloring and to colour v_0 with a new colour. In other words, we have

$$\chi(G_1^c) \leqslant (\chi_1 + 1) \tag{4}$$

Based on the same reasoning using $(G_2 \cup G(F_1))^c$, instead, we have

$$\chi((G_2 \cup G(F_1))^c) \leqslant (\chi_2 + 1) \tag{5}$$

Using (1) and (5), we obtain the following as required:
$\max(\chi_1, \chi_2) \leqslant \chi(G) \leqslant \max(\chi_1, \chi_2) + 1$

□

Suppose now there exist a χ_1 colouration of graph G_1 (resp χ_2 colouration of the graph G_2) and then let $S^1 = \{S_1^1, S_2^1, ..., S_{\chi_1}^1\}$ be a partitioning of graph G_1 into stable set (resp $S^2 = \{S_1^2, S_2^2, ..., S_{\chi_2}^2\}$ of a graph G_2).
G_1^c (respectively G_2^c) be a graph obtained after the contraction of the sets $\{S_1^1, S_2^1, ..., S_{\chi_1}^1\}$ of the graph G_1 (resp $\{S_1^2, S_2^2, ..., S_{\chi_2}^2\}$ of a graph G_2) into vertices $v_1^1, v_2^1, ..., v_{\chi_1}^1$ (resp $v_1^2, v_2^2, ..., v_{\chi_2}^2$)
$G^c = G_1^c \cup G_2^c$
$F_1^c = \{v \in V(G_1^c)/\exists v^{'} \in V(G_2^c) : vv^{'} \in E(G^c)\}$
$F_2^c = \{v \in V(G_2^c)/\exists v^{'} \in V(G_1^c) : vv^{'} \in E(G^c)\}$
and $F^c = G(F_1^c \cup F_1^c)$ be a sybgraph induced by the vertices in $F_1^C \cup F_2^C$.

Property 3.3. $\chi(G^c) = \max(\chi_1, \chi_2, \alpha(\overline{F}^c))$.

Proof. The graph G_1^c which is obtained after the contraction of the sets $\{S_1^1, S_2^1, ..., S_{\chi_1}^1\}$ of the graph G_1 into vertices $v_1^1, v_2^1, ..., v_{\chi_1}^1$ is a clique. This can be shown by contradiction.
Asuume that the opposite is true:
It would exist two vertices v_i^1 and v_j^1 $i, j \in \{1, 2, ..., \chi_1\}$ such that $v_i^1 v_j^1 \notin E(G_1^c)$: then $\forall v \in S_i^1$, $\forall v^{'} \in S_j^1 : vv^{'} \notin E(G_1)$ so the new set $S_i^1 \cup S_j^1$ is a stable set and therefore G_1 can be coloured with $(\chi_1 - 1)$ colours. This is contradiction that implies that G_1^c is a clique. The same proof can be used for the graph G_2
According to property 1, we shall therefore have:
$\chi(G^c) = \max(\chi_1, \chi_2, \alpha(\overline{F}^c))$. □

Property 3.4. *If* $\chi(G^c) = \max(\chi_1, \chi_2)$ *then* $\chi(G) = \max(\chi_1, \chi_2)$, *otherwise (i.e:* $\chi(G^c) \neq \max(\chi_1, \chi_2)$*)* $\max(\chi_1, \chi_2) \leqslant \chi(G) \leqslant \alpha(\overline{F^c}) \leqslant \chi_1 + \chi_2$

Proof. It is well known that $\max(\chi_1, \chi_2) \leqslant \chi(G) \leqslant \chi(G^c)$ then if $\chi(G^c) = \max(\chi_1, \chi_2)$ it is obvious that $\chi(G) = \max(\chi_1, \chi_2)$.
Otherwise, $\chi(G^c) \neq \max(\chi_1, \chi_2)$ and according to the property 3, $\chi(G^c) = \alpha(\overline{F^c})$ then $\max(\chi_1, \chi_2) \leqslant \chi(G) \leqslant \alpha(\overline{F^c})$. However, $\alpha(\overline{F^c})$ is a cardinal of the maximum stable set of the bipartite graph $\overline{F^c} = \overline{F_1^c \cup F_2^c}$ then $\alpha(\overline{F^c}) \leqslant \mid F_1^c \mid + \mid F_2^c \mid$. So $\mid F_1^c \mid \leqslant \chi_1$. (resp $\mid F_2^c \leqslant \chi_2 \mid$) because the vertices of F_1^c (respectively F_2^c) represente the colours of F_1 (respectively of F_2) then $\alpha(\overline{F^c}) \leqslant \mid F_1^c \mid + \mid F_2^c \mid \leqslant \chi_1 + \chi_2$. This completes our proof i.e:
$\max(\chi_1, \chi_2) \leqslant \chi(G) \leqslant \alpha(\overline{F^c}) \leqslant \chi_1 + \chi_2$.
□

This property has the advantages of providing both lower and upper bounds of the chromatic number of a graph G. This invaluable information can obviously be incorporated within heuristic search methods to seep up the search and hence determine optimal or near optimal values for $\chi(G)$ more efficiently.

4 Applications

In this section we describe our method for some classes of graphs and some instances DIMACS data sets. Note that coloring graphs that fit in the first class can be performed in polynomial time, see for instance the review chapter by Maffray (2003).

4.1 Chordal Graphs

A graph G is triangulated if every cycle of length greater than three has an edge joining two nonadjacent vertices of the cycle. The edge is called achord,and triangulated graphs are also called *chordal graphs*, see Hajnal and Suranyi (1958). A vertex u of a graph G is a simplicial vertex if and only if the induced sub graph of $\mathcal{N}(u)$ is a clique. A perfect vertex elemination scheme of a graph G is an ordering $u_1, u_2, ..., u_n$ such that u_i is a simplicial vertex for all $i = 1, ..., n-1$ of the subgraph of G induced by $u_1, u_2, ..., u_n$. It is also called a perfect scheme. If a graph G is chordal, according to the theorem of Tarjan and Yannakakis (1984) which states that *A graph G is chordal if and only if it exists a perfect vertex elimination scheme of a graph G*, there does therefore exist an ordering $u_1, u_2, ..., u_n$ which is a perfect vertex elimination scheme of a graph. Let $h(u_i)$ be a set of neighbours of vertex u_i in the subgraph of G induced by $u_1, u_2, ..., u_n$.

Property 4.1. *If a graph G is chordal then:* $\chi(G) = \max(\mid h(u_1) \mid +1, \mid h(u_2) \mid +1, ..., \mid h(u_{|V(G)|-1}) \mid +1)$

Proof. Since G is a chordal graph then according to the theorem of Tarjan and Yannakakis (1984), there exists an ordering $u_1, u_2, ..., u_{|V(G)|}$ which is a perfect vertex elimination scheme of a graph G. Let $h(u_i)$ be a set of neighbours of vertex u_i in the subgraph of G induced by $u_1, u_2, ..., u_n$.
We use a decomosition of the graph G as mentioned in figure 4.
To determine the chromatic number for the graph G. we strat the procedure from the end of the decomposition and recursively we back track till we reach the beginning and hence find the chromatic number of graph G. We know that $u_{|V(G)|-1}$ is a simplicial vertex in the subgraph of G induced by $\{u_{|V(G)|-1}, u_{|V(G)|}\}$ so $\mid h(u_{|V(G)|-1}) \mid$ is a clique in the graph $G(\{u_{|V(G)|-1}, u_{|V(G)|}\})$.
Using property 1 we find that
$\chi(G(\{u_{|V(G)|-1}, u_{|V(G)|}\})) = \max(\chi(G(\{u_{|V(G)|-1}\}), \chi(\{u_{|V(G)|}\}), \alpha(\overline{F_{|V(G)|}})) =$ $\max(1, 1, \alpha(\overline{F_{|V(G)|}})$ where $F = (G(h(u_{|V(G)|-1})) \cup \{u_{|V(G)|} - 1\})$ since $\alpha(\overline{F_{|V(G)-1|}}) =\mid h(u_{|V(G)|-1}) \mid +1$ $(F = (G(h(u_{|V(G)|-1})) \cup \{u_{|V(G)|} - 1\}))$ is a clique so
$\chi(G(\{u_{|V(G)|-1}, u_{|V(G)|}\}) =\mid h(u_{|V(G)|-1}) \mid +1$.
Recursively for every $i \in \{1, 2, ..., \mid V(G) \mid -1\}$, u_i is a clique in the subgraph of G induced by $u_1, u_2, ..., u_{|V(G)|}$ so $h(u_i)$ is a clique in the graph $G(u_1, u_2, ..., u_{|V(G)|})$
Using the property 1 we find that
$\chi(G(u_i, u_{i+1}, ..., u_{|V(G)|})) = \max(1, u_{i+1}, u_{i+2}, ..., u_{|V(G)|})$, $\alpha(\overline{F}_i)$ where $F_i = G(h(u_i) \cup \{u_i\})$ since $\alpha(\overline{F}_i) =\mid h(u_i) \mid +1$
$(\ F_i = G(h(u_i) \cup \{u_i\}))$ is a clique in the graph
$G(u_i, u_{i+1}, ..., u_{|V(G)|})$ and
$\chi(G(u_{i+1}, u_{i+2}, ..., u_{|V(G)|})) = \max\{\mid h(u_{i+1}) \mid +1, \mid h(u_{i+2}) \mid +1, ..., \mid h(u_{|V(G)|-1}) \mid$

$+1\}$ (by the procedure of recursivity).
Hence, $\chi(G(u_i, u_{i+1}, ..., u_{|V(G)|})) = \max(\mid h(u_i) \mid +1, \mid h(u_{i+1}) \mid +1, ..., \mid h(u_{|V(G)|-1}) \mid +1)$
In particular, for $i = 1$, $\chi(G(u_1, u_2, ..., u_{|V(G)|})) = \max(\mid h(u_1) \mid +1, \mid h(u_2) \mid +1, ..., \mid h(u_{|V(G)|-1}) \mid +1)$
So $\chi(G) = \max(\mid h(u_1) \mid +1, \mid h(u_2) \mid +1, ..., \mid h(u_{|V(G)|-1}) \mid +1)$.

□

4.2 Weakly Triangulated Graphs

4.3 Split Graphs

4.4 Instance DIMACS

5 Graph Coloring via Branch and Price

Min-GCP can be seen as the problem of partitioning the vertices of a graph into the minimum number of stable sets, since all stable set vertices can be assigned to the same color, no matter which, Being a minimiszation problem, MIN-GCP can be formulated as a set covering where the ground set is given by V and the family of subsets is a collection of maximal stable sets of G. Alternatively, a set packing formulation of MIN-GCP is presented in Hansen et al.(2009), but since it yields the same lower bound than the set covering formulation.
Let S be the collection of all maximal stable of G = (V,E). Denoting by x_s the selection variable of stable set s, the set covering formulation of MIN-GCP is:

$$Z_{MP} = \min \sum_{s} x_s \tag{6}$$

$$\sum_{s:i \in S} x_s \geq 1 \qquad \forall i \in V \tag{7}$$

$$x_s \in \{0, 1\} \qquad \forall s \in S \tag{8}$$

In Column Generation scheme, we consider the so called restricted master problem :

$$Z_{MP} = \min \sum_{s} x_s \tag{9}$$

$$\sum_{s:i \in \overline{S}} x_s \geq 1 \qquad \forall i \in V \tag{10}$$

$$x_s \in [0, 1] \qquad \forall s \in \overline{S} \tag{11}$$

where $\overline{S} \subseteq S$ is such that the existence of a feasible solution is guaranteed, that is $\overline{S}$ is a subset of maximal stable sets such that each vertex i of G is contained in at least one set of $\overline{S}$. Let $\overline{Z_{MP}}$ denote the linear relaxation optimal solution value of problem (6)-(8).Note that Z_{MP} is equal to the chromatic number $\chi(G)$, and that

Let $\overline{Z_{MP}}$ is equal to the fractional chromatic number $\chi_f(G)$. The computation of $\chi_f(G)$ is an important **NP**-hard problem, and it gives the sharpest known lower bound for χ (Schrjiver, 2008).

Consider the dual of problem (9)-(11) where we denote by π_i the dual variables of constraints (10), given the optimal dual solution π_i, to check the existance of negative reduced columns to be added to (9)-(11), the following Pricing subproblem has to be solved:

$$s^* = \max \sum_{i \in V} \pi_i z_i \tag{12}$$

$$z_i + z_j \leqslant 1 \qquad \forall (i,j) \in E \tag{13}$$

$$z_i \in \{0,1\} \qquad \forall i \in V \tag{14}$$

Note that this problem is equivalent to finding the maximum weight stable set: variable z_i is equal to one if vertex i is part of a stable set and constraint (13) avoid that neighbors of i are in the solution. Negative reduced cost columns correspond to solutions of problem (12)-(14) with value strictly greater than 1. thus the actual value of the pricing subproblem is $c^* = 1 - s^*$

6 A New Model for the decomposition of Graph

We consider a new formulation for MIN-GCP which can be obtained from the last formulation of the set covering problem by using a suitable decomposition scheme. Let v the partitioning of the vertices of V .We create a formulation with binary variables x_s for each $s \in v$ and $s \subset V$. $x_s = 1$ implies that the set s will be given a unique label, while $x_s = 0$ implies that the set does not require a label. The MIN-GCP is then the following:

$$Z^{'}_{LMP} = \min \sum_{s \subseteq V} \overline{\chi}(G(s)) x_s \tag{15}$$

$$\sum_{s \subseteq V} a_s^v x_s \geq 1 \qquad \forall v \in V \tag{16}$$

$$x_s \in \{0,1\} \qquad \forall s \subseteq V \tag{17}$$

where $\overline{\chi}(G(s))$ is the upper bound of $\chi(s)$. Since this formulation has also one constraint for each vertex, but can have a tremendous number of varaibles. Note that a feasible solution of this problem may assign multiple labels to a vertex. This can be corrected by using any one of the multiple labels as the label for the vertex.

The fact remains, However, this formulation can have far more variables than can be reasonably attacked directely, we use then the well known technique so called *Column Generation* to solve the linear relaxation of this problem. Let λ_v^* v be the value of the reduced cost of the constraint (16) associate to a vertex $v \in V$. The pricing subproblem take then to find the set $s \in V$ where

$$\sum_{v \in s} a_s^v \lambda_v^* > \chi(G(s)) \tag{18}$$

6.1 Branching Rule

A difficult part about using column generation for integer programs is the development of branching rules to ensure integrality. Rules that are appropiate for integer programs where the entire set of columns is explicitly available do not fit in well with restricted integer programs where the columns are generated by implicit techniques.[2] Consider, for instance, the rule of branching on a fractional variable, where the variable is set to 1 in one subproblem and to 0 in the other. The former subproblem causes no problem for (LMP): setting a set variable s_i to 1 corresponds to applying a $\chi(G(s_i))$ lable "colors" to those vertices.Those vertices can then removed from the graph. The other problem is more difficult. Setting a variable to 0 corresponds to not permitting the use of that set. How can this information be passed to the subproblem(that generates partition sets) Our approach to the integrality problem is to use a branch and bound method without increasing the complexity of the subproblems. We accomplish this by devising special branching rules that ensure that the subproblem to be solved for each branch is itself a graph coloring problem without any additional constraints and can be solved by our column generation methodology.

Zykov Branching scheme: Given a graph G and two non-adjacent vertices u and w, consider the following two operations:*contraction and insertion*.
the contraction (or shrinking) operation, denote by $G\backslash\{u, w\}$, consists of creating a new vertex v with an incident edge for each vertex in the union of the neighborhoods of u and w and of removing vertices u and w from the vertex set. The insertion operation consists of adding the edges $\{u, v\}$ to E, and is denoted by $G + \{u, v\}$.

Theorem 6.1. $\chi(G) = \min\{\chi(G\backslash\{u, w\}), \chi(G + \{u, v\})\}$

This well-known result has been exploited in a number of coloring algorithms that are usually called Zykov's (*contraction*) algorithms.[3] The branching scheme derived from this result.

6.2 Solving the Pricing SubProblem

We propose an exact and two heuristics algorithm for the solution of the pricing subproblem. The exact algorithm consists in solving the model, eventually augmented by branching constraints, through a commercial solver. On the heuristics side, we use a **tabu** search algorithm since it has been shown to be very effictive in finding optimal satble sets and clique sets.

6.3 Column Generation

The set $\overline{S}$ used to initialize the RLMP is constructed as follows. At the root of the branch-and-bound tree, we first generate k sets $s_1, s_2, ..., s_k$. Each s_i is initialized to empty set, and the vertices of G are then considered, one after the other, for possible inclusion in a set s_i.Since there are two ways to find the set(or the subsets) s_i. Indeed, we can just look for a new subgraph in G or simply take two subsets s_i,s_j already generated and then concatenate. For the first problem it is possible to generate two times the same s with $\overline{\chi}$ different using the concatenation, in this case we keep the best and we remove the other. Note that it may happen that $\bigcup_{i=1}^{k} s_i \neq V$, which means that the RLMP has no feasible solution if restricted to this subset $\overline{S}$. For this reason, at each node of the branch-and-bound tree, we add a high cost dummy column in $\overline{S}$. It represents a fictitious color class containing all vertices of G. This column is kept in the model until feasiblity is reached. Also, at each node of the branch-and-bound tree, all columns generated so far which are feasible with respect to branching constraints are inserted in the RLMP. Once the RLMP has been solved to optimality, we consider the pricing subproblem in order to find new negative reduced cost columns. If the optimal solution of the pricing subproblem has a positive value, it means that the op- timal solution of the current RLMP is also optimal for the LMP. Algorithm 1 summarizes the column generation algorithm and its implementation follows the scheme given in Figure 5.

7 Computational Results

In this section we describe our computational experience with BP i.e. the branch and price algorithme implementation in this study. Experiments were carried on a Sun ULTRA1 workstation with CPU running at 140 MHz and 288 Mb of RAM memory. CPU times are reported in seconds.

DIMACS benchmark instances were used in the experiments.we can generate some random instances

Table 1 describe the DIMACS instances and the number of vertices, number of edges and the size of maximal cliques are given. The rightmost table column indicates the chromatic number of the corresponding graphs ("?" stands for unknown).

problem	vertices	Edges	ω	BP	BPD
DSJC125.1	125	736	3	5	
DSJC125.5	125	3891	9	?	%
DSJC125.9	125	6961	28	?	%
DSJC250.1	250	3218	4	?	%
DSJC250.5	250	15668	10	?	%
DSJC2500.9	250	27897	34	?	%
DSJC500.1	500	12458	4	?	%
DSJC500.5	500	62624	10	?	%
DSJC500.9	500	112437	44	?	%
DSJC1000.9	1000	49629	5	?	%
DSJC1000.1	1000	249826	13	?	%
DSJC1000.5	1000	449449	51	?	%
DSJR500.1	500	3555	11	12	%
DSJR500.1C	500	121275	67	85	%
DSJR500.5	500	58862	111	122	%
latin-sq-10	900	307350	90	?	%
le450-5a	450	5714	5	5	%
le450-5b	450	5734	5	5	%
le450-5c	450	9803	5	5	%
le450-5d	450	9757	5	5	%
le450-15a	450	8168	15	15	%
le450-15b	450	8169		15	%
le450-15c	450	16680		15	%
le450-15d	450	16750		15	%
le450-25a	450	8260		25	%
le450-25b	450	8263		25	%
le450-25c	450	17343		25	%
queen5-5	25	160		5	%
queen6-6	36	290		7	%
queen7-7	49	476		7	%
queen8-8	64	728		9	%
queen8-12	96	1368		12	%
queen9-9	81	1056		10	%
queen10-10	100	2940		11	%
queen12-12	121	3960		11	%
queen13-13	144	5192		12	%
queen14-14	196	8372		14	%
queen15-15	225	10360		15	%
queen16-16	256	12640		16	%
myciel2	5	5		3	%
myciel3	11	23		4	%
myciel4	20	71		5	%
myciel5	47	236		6	%
myciel6	95	755		7	%
myciel7	191	2360		8	%

8 Conclusion

In this paper we propose a set covering formulation with an exponential number of variables for graph coloring problem. The model is solved through a column generation algorithm embedded in a branch-and-bound scheme, giving rise to branch -and-price solution approach. In order to decrease the solution time, a tabu seach algorithm is proposed to solve the pricing subproblem. The exact pricing solver is called only when the tabu search is not able to find any negative reduced sost column. talk about the computational experiments

References

[1] Federico Malucelli Stefano Gualandi. Exact solution of graph coloring problem via constraint programming and column generation. *milano*, 2000.