UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

Packing coloring : study of mixed integers formulations

Romain Vandemaele

Promoteur :
Prof. Bernard Fortz

Travail préparatoire au mémoire
présenté en vue de
l'obtention du grade de
Master en Sciences Informatiques

Année académique 2015 - 2016

“*Science is what we understand well enough to explain to a computer, Art is all the rest.*”

Donald E. Knuth[1], 1996

“*The purpose of computation is insight, not numbers.*”

Richhard Hamming[2], 1962

[1] http://www-cs-faculty.stanford.edu/~uno/

[2] https://en.wikipedia.org/wiki/Richard_Hamming

Contents

List of Figures

Glossary

C_n Cycle graph on n vertices. ii

E The set of edges. ii

P_n Path graph on n vertices. ii

Q_n Hypercube graph on vectors of size n. ii

V The set of vertices. ii

$\chi(G)$ The minimum number of color needed for a valid coloring of G. ii

$\chi_P(G)$ The minimum number of color needed for a valid packing coloring of G. ii

$\mathbb{B}$ The set $\{0, 1\}$. ii

$\omega(G)$ The maximum size of a clique of G. ii

$\sigma(G)$ The maximum size of an independent set of G. ii

$\tau(G)$ The minimum size of a vertex cover of G. ii

m Number of edges in a graph. ii

n Number of vertices in a graph. ii

s.t. such that. ii

Acknowledgements

I like to acknowledge my advisor Bernard Fortz for its advices.

Chapter 1

Introduction

1.1 Notions

In this document, every graph is an undirected graph and n and m denote respectively the number of vertices($|V|$) and the number of edges($|E|$). Moreover an edge between two vertices u and v is denoted uv.

Definition 1. ***The distance between u and v*** *in a graph G=(V,E) with* $u, v \in V$ *noted as* d_{uv} *is the length of the shortest path from* u *to* v*. Two vertices are* ***adjacent*** *iff the distance between them is one.*

Definition 2. ***An induced subgraph*** *of* $G = (V,E)$ *is a graph* $H = (V', E')$ *s.t.* $V' \subseteq V$ *and* $uv \in E' \iff u, v \in V' \ \wedge \ uv \in E$*. H is denoted as a subgraph of G induced by* V'*. H is induced by* E' *if* $E' \subseteq E \ \wedge \ u \in V' \iff \exists v \in V \ uv \in E'$

Definition 3. ***A clique*** *is a subgraph of G=(V,E) induced by* $S \subseteq V$ *s.t.* $\forall u, v \in S$ *with* $u \neq v, uv \in E$*. The maximum size of a clique in G is denoted here by* $\omega(G)$ *and referred as the* ***clique*** *number. If* $S = V$ *then G is called a clique and denoted* K_n*.*

Definition 4. ***An independent set*** *is an subgraph of G=(V,E) induced by* $S \subseteq V$ *s.t.* $\forall u, v \in S$ *with* $u \neq v, uv \notin E$*. The maximum size of a independent set in G is denoted here by* $\sigma(G)$ *and referred as the* ***independence number****.*

Definition 5. ***A vertex cover*** *of a graph G=(V,E) is a subset* $C \subseteq V$ *s.t.* $\forall uv \in E, u \in C \vee v \in C$*. The maximum size of a vertex cover in G is denoted here by* $\tau(G)$*.*

1.2 Problem

The packing coloring is a special case of the famous coloring problem. This problem consists of assigning a color to each vertex of a graph s.t. two adjacent vertices cannot have the same color. Such a coloring is called **valid**. A valid coloring of a graph is showed in Figure 1.1. Here it is enough to alternate between 2 colors when coloring them in order. This coloring is valid because each vertex is only adjacent

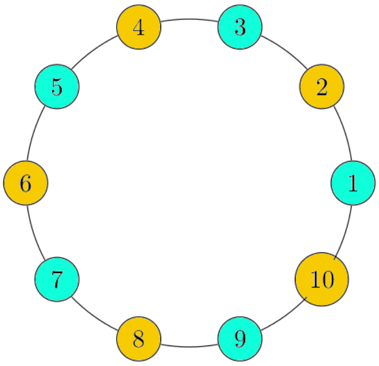

Figure 1.1: Example of a graph's coloring

to the previous and next vertex modulo 10 and the number of vertices is even so the last vertex colored has not the same color as the first one.

The packing coloring problem considered here, consists also of a coloring but the constraints are different. Indeed, two vertices having the same color i must have a distance of at least $i + 1$. For instance, Figure 1.2 presents a valid packing coloring of a graph with this correspondence between colors and their number in Table 1.1 :

In this case, more colors are needed than with the coloring by using twice the minimal number of colors to obtain a valid coloring. On both figures, only one valid coloring is shown but there can have a large amount of them even with the minimal number of colors. For instance, the colors can be rotated s.t. vertex i has the color of vertex $i + jmodn$ with $j \in \mathbb{Z}$.

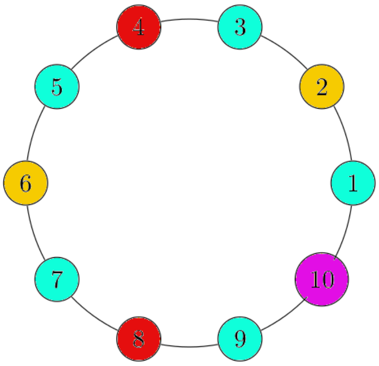

Figure 1.2: Example of a graph's packing coloring

Number	1	2	3	4
Color[1]	Bright turquoise	Yellow (Munsell)	KU Crimson	Phlox

Table 1.1: The correspondence of colors in Figure1.2.

1.3 Description

Now that the subject is introduced, the next sections enter in greater details about colorings. The most important results and formulations of the coloring problem are presented in Section 2. The Section 3 follows the same structure by extending results of section 2 if possible but specific results about this problem are also presented.

Chapter 2

Coloring

2.1 Description

The problem was briefly described in Section 1, the goal is to find a function f that assigns colors to every vertex of a graph $G = (V,E)$ defined as follow :

$$\begin{aligned} f : V &\longrightarrow C \\ f(u) \neq f(v) \;\forall\; uv &\in E \\ C &\subseteq \mathbb{N} \end{aligned} \tag{2.1}$$

There is two main ways to see the problem. On one hand, a decision problem which is a problem where the response is either yes either no. The question is whether G has a valid coloring with k color or not. For this question k is either an input as G or is fixed which influences its complexity. If k is fixed then it is enough to test all nk possible coloring which is polynomial in the input size. This problem defines a set of languages $L = \{L_i \;\forall i \in \mathbb{R}\}$ with $L_k = \{< G >\text{: G has a valid coloring with k colors}\}$ with $< G >$ a binary encoding of G.
On the other hand, an optimization problem where the goal is to minimize or maximize an objective function g under some constraints on variables. For example, $|C|$ has to be minimized also called the chromatic number under the constraint that the coloring is valid. Some concrete examples are described in Section 2.3.

Definition 6. ***The chromatic number*** *of a graph G, denoted $\chi(G)$, is the minimum number of colors needed for a valid coloring of a graph G.*

Definition 7. ***A planar graph*** *is a graph that can be drawn s.t. there is no crossings between the edges and between a vertex and an edge. The graph has to be drawn s.t. each vertex is represented as a unique point in $\mathbb{R}^2$ and edges are represented as lines between the 2 corresponding points.*

This problem has many applications like the coloration of a map s.t. two neighboring countries have different colors to be able to distinguish the frontiers. In

1977, Kenneth Appel and Wolfgang Haken[1] proved first the four color Theorem in [1] that states that for any planar graph G $\chi(G) \leq 4$.

2.2 Results

There are several results on this very popular problem, the most general ones are presented here.

2.2.1 Complexity

Theorem 1. *The coloring problem is NP-complete*

Proof. The decision problem belongs to the NP class. Indeed, a poly-time verifier would take the neighborhood of every vertex and test if the constraint is respected. This verifier would be in $\mathcal{O}(n+m)$. For the NP-hard property, a proof by reduction from $3-SAT$ is in [14][2]. □

Theorem 1 implies that no polynomial algorithm exists until proven otherwise[3].

2.2.2 Lower and Upper bounds

Theorem 2. *For any graph* $G = (V,E)$, $\omega(G) \leq \chi(G) \leq n$.

Proof. For the upper bound, it is easy to observe that there cannot be more than n colors because each vertex is associated with one and only one color.
The lower bound is proved by contradiction. Suppose $\chi(G) < \omega(G)$ then two vertices u and v in a clique of size $\omega(G)$ have the same color by the pigeonhole principle but by the definition of a clique u and v are adjacent. □

This is very general bounds it can be reduced by taking independent set into account because every vertex in an independent set can be colored by the same color without breaking any constraints. Thereby the upper bound can be reduced as stated in Theorem 3.

Theorem 3. *For any graph* $G = (V,E)$, $\chi(G) \leq |S| + n - \sum_{s \in S} |s|$.

$S = \{s_1, s_2, \ldots, s_n | s_i \text{ is an independent set } \forall i \in \{1, \ldots, n\} \wedge s_i \cap s_j = \emptyset \ \forall i,j \in \{1, \ldots, n\} \ i \neq j\}$

We take S *having the maximal* $\sum_{s \in S} |s|$ *among all set having this property.*

Proof. All set of S are independent so each one can be colored with a unique color thus it uses $|S|$ color at most. We still have to color $n - \sum_{s \in S} |s|$ vertices. In the worst case, each of them will have one unique color.

□

[1] A shorter proof was made in [12]
[2] In the version with exercise correction
[3] P = NP ?

The upper bound can still be reduced by using the vertex cover number ,$\tau(G)$ by observing its links with independent set :

Theorem 4. *(Gallai's) For any graph $G = (V,E)$, $n - \sigma(G) = \tau(G)$.*

Proof. Let C be a minimum vertex cover, the set $I = V \setminus C$ is an independent set. If $|I| \leq 1$ then it is trivial. Otherwise, by contradiction suppose that there exists $a, b \in I$ that are adjacent and let e the edge between a and b. Then e is not covered by C which contradicts our hypothesis that V is a vertex cover. By the same idea, the complement of an independent set is a vertex cover.
We still have to prove that I is a maximum independent set. By contradiction, let $I\prime$ be an independent set s.t. $|I| < |I\prime|$ then $C\prime = V \setminus I\prime$ is a vertex cover with $|C\prime| < |C|$ but C is a minimal vertex cover. □

By Theorem 4 and Theorem 3, corollary 1 can be proved easily.

Corollary 1. *For any graph $G = (V, E)$, $\chi(G) \leq \tau(G)+1$.*

2.3 Formulations

This sections presents several formulation of the optimization problem. There exists several formulations with distinct number of variables and constraints.

2.4 Vertex color

This first formulation is the more intuitive one, there is a boolean variable x_{ic} for every vertex i and color c thereby the number of variables is in $\mathcal{O}(n^2)$ because there are at most n colors as proved in Theorem 2.

$$\forall i \in V \forall c \in C \ x_{ic} = \begin{cases} 1 & \text{if the color of vertex i is c} \\ 0 & \text{otherwise} \end{cases}$$

In addition to the constraint on adjacent vertices, the fact that a vertex must have exactly one color must be ensured. This gives the following formulation :

$$\begin{aligned}
&\textit{minimize} && z && \quad (2.2)\\
&\textit{subject to} && \sum_{c=0}^{n} x_{ic} = 1 && \forall i \in V \quad (2.3)\\
& && x_{ic} + x_{jc} \leq 1 && \forall ij \in E,\ \forall c \in C \quad (2.4)\\
& && x_{ic}c \leq z && \forall c \in C,\ \forall i \in V \quad (2.5)\\
& && x_{ic} \in \mathbb{B} && \forall i \in V,\ \forall c \in C \quad (2.6)\\
& && z \in \mathbb{Z} && \quad (2.7)
\end{aligned}$$

In the objective function 2.2, z represent the number of color used which is the number to minimize. Constraint 2.3 implies that each vertex has one and exactly one color. Constraint 2.4 implies the condition on adjacent vertices because if they have the same color the sum would be 2. Finally, constraint 2.5 implies that if color $c \in \mathbb{N}$ is used then z is at least c. In other words, z cannot be lower than the larger color used and cannot be greater because it is minimized so z is well-defined.

Definition 8. ***The density*** *of a graph G=(V,E) is the number of edges compared to the maximum number of edges possible.*

$$density = \frac{2|E|}{|V|(|V|-1)}$$

The number of constraints is $\mathcal{O}(n + nm + n^2) = \begin{cases} \mathcal{O}(nm) & \text{if density } > \frac{2}{n} \\ \mathcal{O}(n^2) & \text{otherwise} \end{cases}$

A problem is that this formulation is **symmetric** which means that several solutions are equivalent. Suppose we have a graph which is a clique of size 5. The two results $z = 5$, $x_{11} = 1$, $x_{22} = 1$, $x_{33} = 1$, $x_{44} = 1$, $x_{55} = 1$ and $z = 5$, $x_{11} = 1$, $x_{22} = 1$, $x_{33} = 1$, $x_{45} = 1$, $x_{54} = 1$,represented only by the variables which are true, are equivalent. Indeed, each vertex has exactly one unique color in both solution. This may cause problems when resolving it by branch and bound which is a technique to reduce the set of solutions to explore by applying bounds. This technique implies imposing values to variables to reduce possibilities for the others but forcing a color on a vertex here could lead to an equivalent solution.

2.5 Independent set

Definition 9. *A* ***partition*** *of a set A is a set $P = \{A_1, A_2, \dots, A_n | A_i \subseteq A \ \forall i \in \{1, 2, \dots, n\}\}$ of subsets s.t. $A_i \cap A_j = \emptyset \ \forall i, j \in \{1, 2, \dots, n\}$ $i \neq j$ and $\bigcup_{i \in \{1,2,\dots,n\}} A_i = A$.*

Firstly, we can observe that a coloring creates a partition of V with $P_1, P_2, \dots, P_\chi$ as subsets s.t. each vertex has the same color as every vertices that are in the same subset and has a different color than every vertex that is in a different subset. Moreover, a vertex has one color so is in exactly one subset. Each subset is an independent set because there cannot be any edges between two vertices having the same color.

From this observation, a formulation can express the fact that an independent set is taken as a partition. Let I be the set of independent set, the variable x is defined as follows :

$$\forall i \in I \ x_i = \begin{cases} 1 & \text{if the independent set i is taken as a partition} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} & \textit{minimize} && \sum_{i \in I} x_i && (2.8) \\ & \textit{subject to} && \sum_{i \in I | v \in i} x_i = 1 \quad \forall v \in V && (2.9) \\ & && x_i \in \mathbb{B} \qquad \forall i \in I && (2.10) \end{aligned}$$

In the objective function 2.8 the number of partitions is minimized which is equal to the number of colors. The constraint 2.9 states that the sum of x_i for every independent set i containing a vertex v is one thereby a vertex is in one and only one chosen independent set. Compared to 2.4, there are less constraints more precisely n but far more variables. Indeed, there is $\mathcal{O}(2^n)$ variables because in the worst case, every subset is an independent set.

This number can be reduced by taking into account only the independent set which are maximal which implies that no vertex can be added to them without keeping them an independent set. Suppose that in the solution there is an independent set S which is not maximal. Thereby, there exists a vertex v that can be added to S and keep it independent. The new set is denoted $S\prime = S \cup \{v\}$. The solution can be modified to take $S\prime$ instead of S and replace T the independent set taken that contains v by $T\prime = T - v$. We can do that because $S\prime$ and $T\prime$ were not taken in the original solution because it will not respect constraint 2.9.

Its exponential number of variables is an important drawback but some technique like column generation can facilitate the computations. Indeed, a column generation or branch and price algorithm for the multi-coloring problem is described in [11]. In this another variant, each vertex is associated to a set of colors. Two sets of two adjacent vertices must have an empty intersection. The case when the set has a cardinality of one is the classical coloring thus multi-coloring is a generalization of the famous problem. Another inconvenient of this formulation is that finding an independent set is also NP-complete(cf. [14] for a proof).

2.6 Representative vertex

There exists another formulation with less variables than 2.5 and conserving the same amount of constraints of 2.4. As its name implies, there are representative which are vertices that represent a color. There is only one representative per color. From this idea, the variables x is derived :

$$\forall u, v \in V \ x_{uv} = \begin{cases} 1 & \text{if } u \text{ represents the color of } v \\ 0 & \text{otherwise} \end{cases}$$

A coloration is derived in 2 steps. First, find all representatives $u_1, u_2, \ldots, u_\chi \in V$ s.t. $x_{uu} = 1$ and assign them a different color. Finally, the color of $v \in V$ is the color of its representative vertex $u_i \in V$ s.t. $x_{u_i v} = 1$. The formulation is as follow

$$\begin{aligned}
&\textit{minimize} && \sum_{u \in V} x_{uu} && && (2.11)\\
&\textit{subject to} && \sum_{u \in V} x_{uv} = 1 && \forall v \in V && (2.12)\\
& && x_{uv} + x_{uv\prime} \leq x_{uu} && \forall u \in V \ \forall vv\prime \in E && (2.13)\\
& && x_{uv} \in \mathbb{B} && \forall u, v \in V && (2.14)
\end{aligned}$$

The objective function 2.11 minimizes the number representatives thus also the number of colors. Constraint 2.12 states that each vertex has one and only one representative and constraint 2.13 implies that two adjacent vertices cannot have the same representative and that if a vertex u represents another one, x_{uu} is forced to one. There is n^2 variables and $n + nm = \mathcal{O}(nm)$ constraints. We have the same performance as 2.4 but the number of constraints can still be reduced in some cases.

Definition 10. ***The neighborhood*** N ***of a vertex*** u *in a graph* $G = (V, E)$ *is the set of vertices that are adjacent to* u*.*

$$N(v) = \{v \in V \setminus \{u\} | uv \in E\}$$

Definition 11. ***The neighborhood*** N ***of*** $S \subseteq V$ *in a graph* $G = (V, E)$ *is the set of vertices that are adjacent to a vertex in* S *but are not in* S*.*

$$N(S) = \{v \in V \setminus S | \exists u \in S \ uv \in E \ \}$$

Definition 12. *The* ***anti-neighborhood*** $\bar{N}(S)$ *for* $S \subseteq V$ *is the set of vertices not accessible from a vertex in* S *via only one edge.*

$$\bar{N}(S) = \{v \in V \setminus S | uv \notin E, \forall u \in S\}$$

The anti-neighborhood can also be defined as the neighborhood in the complementary graph. The anti-neighborhood is very useful for coloring because $\bar{N}(u)$ defines the set of vertices that can be represented by u which reduces the number of constraints. Thereby, we can restricted our variables x_{uv} to all pair $u \in V$ *and* $v \in \bar{N}(u) \cup u$ which gives us $\mathcal{O}(n^2)$ but in practice it is going to be less than n^2 because it's only the case when each vertex has an empty neighborhood. This gives us the following formulation from [6] :

$$\begin{aligned}
&\textit{minimize} && \sum_{u \in V} x_{uu} && && (2.15)\\
&\textit{subject to} && \sum_{u \in \bar{N}(V)} x_{uv} = 1 && \forall v \in V && (2.16)\\
& && x_{uv} + x_{uv\prime} \leq x_{uu} && \forall u \in V \ \forall vv\prime \in E(\bar{N}(u)) && (2.17)\\
& && x_{uv} \in \mathbb{B} && \forall u \in V, \ v \in \bar{N}(u) \cup u && (2.18)
\end{aligned}$$

That gives us $\mathcal{O}(n + n\bar{m})$ with $\bar{m}$ the number of edges in the complementary graph which is better if the original graph has a high density. This formulation like the one in Section 2.4 is symmetric because the solution is the same if only the representatives of some partitions change. For instance, if $u \in V$ represents $v_1, v_2, v_3, \ldots, v_n$ then $\{v_1, v_2, v_3, \ldots, v_n, u\}$ forms an independent set and is also a partition, v_i could become the representative for $v_1, v_2, v_3, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n, u$ and have exactly the same coloring. One solution proposed by [5] is to impose a total order on the vertices and define a deterministic rule to determine the representative of an independent set.

This formulation is often used as base for algorithms on formulations. For example [2] presents a branch and cut algorithms to solve a special case of the coloring problem named equitable coloring. In This variant, the difference in cardinality of two partitions as defined in section 2.5 must be at most one. Figure 1.1 represents an equitable coloring because there is exactly 5 vertices for each color. If a third color was used to recolor only one vertex then the coloring would not be equitable anymore. Indeed, a color would have one vertex and another one would have 4 vertices which makes a difference of 3.

Chapter 3

Packing coloring

3.1 Description

The problem was first introduced by [10] as *broadcast coloring*. The problem was about assigning a frequency to radio antennae. The problem comes from the interference that occurs when two signals with same frequency are too close. There are several signal's power that indicates the minimal distance between two signals with same frequency. There is a correspondence between a frequency and a color. Moreover, the number associated to the color increase with the power of the frequency. Then, several other applications were found like animals' placement. Thereby, this notion was renamed to the one of *packing coloring* by [4]. This come from the notion of **i-packing** which is a set of vertices having the property that the distance between each pair of distinct vertices have a distance of at least $i+1$. For example an independent set is a $1-packing$. As said in Section 1, this problem is also about finding a coloring described by a function f :

$$\begin{aligned} f : V &\longrightarrow C \\ \forall\ uv \in E\ f(u) = f(v) \implies d_{uv} &> f(u) \\ C &\subseteq \mathbb{N} \end{aligned} \tag{3.1}$$

The decision and optimization problems are equivalent to the coloring problem by deciding if a valid packing coloring with k colors exists and minimizing the number of color used. In the same way, the notion of chromatic number is extended to the packing chromatic number denoted $\chi_P(G)$.

3.2 Results

This decision problem has been proved to be NP-complete by [10] except if $k \leq 3$. We denote a valid packing coloring with k colors by a **k-coloring**. Nonetheless, there are several kind of graphs where the packing chromatic number can be bounded. This section is dedicated to present this type of results found in the literature.

3.2.1 General bounds

The bounds presented in Theorem 2 are still correct. Nonetheless $\chi(G)$ and $\chi_P(G)$ are not independent by Theorem 5.

Theorem 5. *For any graph G, $\chi(G) \leq \chi_P(G)$.*

Proof. Every valid packing coloring is a valid coloring thus if a packing coloring with fewer colors than $\chi(G)$ is found then a better coloring has been fond thus contradicting the definition of $\chi_(G)$. □

Definition 13. ***A split graph*** *$G = (V, E)$ is a graph in which V can be partitioned in A and B s.t. A induces a clique and B induces an independent set.*

This is not a strict lower bound so there also exists graphs with $\chi = \chi_P$ like split graphs. In both coloring, each vertex of the partition A has to take an unique color and the ones from B can have the same color which is not used in A except if there is no edge between A and B in which case a color from A van be reused for vertices in B.

3.2.2 Graphs with low packing chromatic number

This section considers graphs that need only a few colors to have a valid packing coloring like path graphs, cycle graphs and star graphs.

Definition 14. ***A path graph*** *denoted P_n is composed of n vertices $v_1, v_2, \ldots, v_n$ s.t. $v_i v_j \in E \Rightarrow j = i + 1$ except if $i = n$.*

Definition 15. ***A cycle graph*** *C_n is composed of only a cycle on n vertices $v_1, v_2, \ldots, v_n$ s.t. $v_i v_j \in E \Rightarrow j = i + 1 mod n$.*

A path graph is very simple and can be colored with only a few of colors as stated by Theorem 6. Theorem 7 states an equivalent proposition for cycle graphs.

Theorem 6. *(by [10]) For any path graph P_n, $\chi_P(P_n) \leq 3$.*

Proof. If $n \geq 4$, [10] found a pattern that results in a valid packing coloring which is

$$121312131\ldots$$

as illustrated in Figure 3.1 This pattern uses only 3 colors. This pattern is optimal because at most $\frac{n}{2}$ vertices can be colored by 1 and at most $\frac{n}{3}$ vertices by 2 with coloring $\frac{n}{3} + \frac{n}{2}$ vertices is impossible because of overlapping. If $n < 4$, the pattern can still be used which results in a 1-coloring or 2-coloring depending on n. □

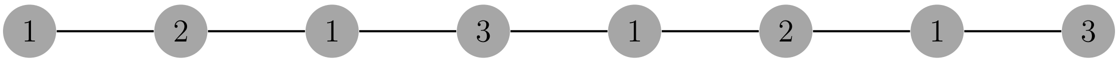

Figure 3.1: Optimal packing coloring of a path graph

A coloring and a packing coloring of a cycle graph are illustrated respectively on Figure 1.1 and 1.2.

Theorem 7. *(by [10]) For any cycle graph C_n, if $n \geq 3$ and n is 3 or a multiple of 4, then $\chi_P(C_n) = 3$; otherwise $\chi_P(C_n) = 4$.*

Proof. (Proof idea) A cycle graph C_n can be seen as P_n in which its two endpoints are linked together with an edge. We could use the pattern described above and then link the endpoints but it can break the coloring. The case $n \leq 3$ being trivial, we consider n as a multiple of 4 then the last vertex completes a pattern thus by linking it to the beginning we create a chain of complete patterns so the coloring is valid as proved by Theorem 6.
In the other cases, some changes have to be done which implies adding a fourth color. In all cases enumerated below, the last vertex is now colored by 4 as illustrated in Figure 1.2 :

- $n = 4r + 1$: 1 2 1 3 . . . 1 2 1 3 1 $\longrightarrow$ 1 2 1 3 . . . 1 2 1 3 **4**
- $n = 4r + 2$: 1 2 1 3 . . . 1 2 1 3 1 2 $\longrightarrow$ 1 2 1 3 . . . 1 2 1 3 1 **4**
- $n = 4r + 2$: 1 2 1 3 . . . 1 2 1 3 1 2 1 $\longrightarrow$ 1 2 1 3 . . . 1 2 1 3 1 2 **4**

□

Definition 16. ***A star graph*** *is a graph denoted S_n that have one central vertex which have $n - 1$ adjacent vertices that have the central vertex as only neighbor. It also is a tree with $n - 1$ leave and with maximum diameter 2.*

At the beginning of Section 3.2, it is mentioned that the decision problem is not NP-hard when $k \leq 3$. This results for $k = 2$ comes from star graphs that are the only graphs with $\chi_P = 2$.

Theorem 8. *(By [10]) For any graph G, $\chi_P(G) = 2 \iff G$ is a star graph.*

Proof. ($\Leftarrow$) A coloring is simply applying the color 2 to the central vertex and 1 to every leaves.
($\Rightarrow$) $\chi_P(G) = 2$ thus P_4 is not a subgraph of G because it needs 3 colors by Theorem 6 thereby $diam(G) \leq 2$. Moreover, the vertex cover number is 1 by corollary 1. It results that the only graphs that have all these properties are star graphs.

□

Definition 17. *A graph G=(V,E) is* ***bipartite*** *if V can be be partitioned in 2 partitions A and B s.t. every edge has one endpoint in A and the other in B.*

Definition 18. *A* ***multigraph*** *is a graph in which there can be several edges sharing the same endpoints.*

The last case to consider is when $k = 3$, every graph G with $\chi_P(G) = 3$ can be created with the following procedure :

1. Start with a bipartite multigraph with bipartition A and B.
2. Subdivide once each edge by adding a vertex between its two endpoints.

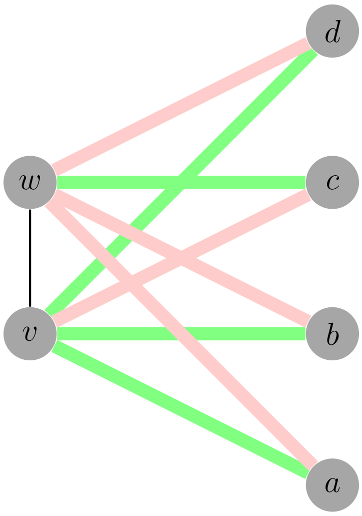

Figure 3.2: Example of a T-add

3. Add leaves in the original set of vertices $A \cup B$.

4. Do a T-add on vertices in B. A T-add is defined in [10]. This is the operation performed on a vertex v which add a vertex w and an independent set X to the graph. Then an edge between v and w is added as well as a subset of all the possible edges between X and $\{v, w\}$. One example is illustrated in Figure 3.2 with $X = \{a, b, c, d\}$. On this figure, green edges are added and red ones are not.

3.2.3 Distance graph

Definition 19. *A* ***distance graph*** *is a graph $G = (V, E)$ and a distance set D with $V \subseteq \mathbb{Z}$ and $D \subseteq \mathbb{Z}^+$ in which each element of V represents a vertex and $E = \{ij | i, j \in V \wedge |i - j| \in D\}$.*

In this section, we consider **infinite distance graphs** which have $\mathbb{Z}$ as vertex set denoted by $D(d_1, d_2, \ldots, d_k)$ with the distance set $D = \{d_1, d_2, \ldots, d_k\}$. The results from the study of the packing coloring number of these graph in [7] are presented in this section. They found several bounds for when $|D| = 2$ as presented :

1. If k, t are coprime (Recall that 2 integers are coprime iff one is their only common divisor) :

   (a) t and k are both odd and $t \geq 825$ then $\chi_P(D(k, t)) \leq 30$.

   (b) k is odd and $t \geq 898$ is even then $\chi_P(D(k, t)) \leq 56$.

   (c) k is even and $t \geq 923$ is odd then $\chi_P(D(k, t)) \leq 56$.

2. If $D(k, t)$ is connected and $t \geq 9$ then $\chi_P(D(k, t)) \geq 12$.

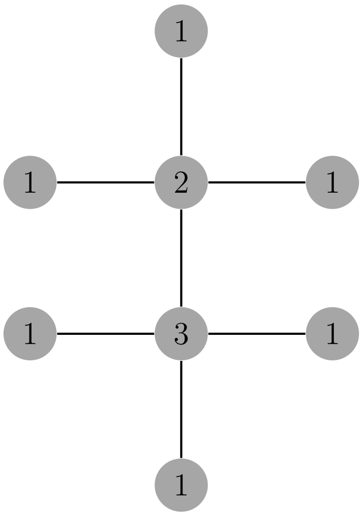

Figure 3.3: Optimal packing coloring of a tree with diameter 3

In addition, they found bounds for graph with small k and t by finding repeatable patterns(upper bounds) or simply by brute-force(lower bounds) with programs they developed. These results are available on the page `http://le2i.cnrs.fr/o.togni/packdist/`. This method is very inefficient because if can take a lot of time. For instance it took them 179 hours to find the lower bounds when $k = 3$ and $t = 7$. A more efficient technique is to use a formulation like they did in [13] with the one presented in Section 3.3. They found new bounds and improved some of those found previously.

3.2.4 Trees

Concerning trees, [10] left as open question to determine if there is a polynomial algorithm to compute χ_P for any tree. [4] suggest that the response was negative by trying two natural approaches but failed. Finally [8] has proved that it is NP-complete even for trees although a classic coloring of a tree is trivial.

Nonetheless, [15] proved that χ_P of trees with maximum degrees 3 is bounded by 7 but also that there is no upper bound once that that maximum degree is above 3. there is a similar situation with diameters as proved by [10].

Indeed, a tree with diameter 3 has a valid packing coloring with 3 colors like illustrated in Figure 3.3. The root can be colored 3, its children 1. Among its child only one vertex v can have children otherwise the distance between two of these children would be 4. The children of v must have color 1 because the distance between them is 2. It creates a problem with v which has also color 1 but v can be recolored with 2. Likewise, there is no bounds for trees with diameter 4 but the packing chromatic number can be computed in polynomial time by result from Theorem 9.

Theorem 9. *Let T be a tree with diameter 4 and a central vertex v. Let $n_i \forall i \in \{1,2,3\}$ denotes the number of neighbors of degree i of v. A vertex is* ***large*** *if its degree is at least 4. Let L be the number of large vertices in the neighborhood of v then :*

$$\text{If } L = 0, \chi_P(T) = \begin{cases} 4 & \text{if } n_3 \geq 2 \text{ and } n_1 + n_3 \geq 3 \\ 3 & \text{otherwise} \end{cases}$$

$$\chi_P(T) = \begin{cases} L+3 & \text{if } n_3 \geq 1 \text{ and } n_1 + n_2 + n_3 \geq 2 \\ L+1 & \text{if } n_1 + n_2 + n_3 = 0 \\ L+2 & \text{otherwise} \end{cases}$$

However, [8] found a polynomial algorithm for trees with n vertices in $\mathcal{O}(n^{2t+3})$ with t a parameter of the S-packing coloring which is a generalization of the packing coloring. In this one the number of each color is determined by a non decreasing sequence $S = (s_1, s_2, \dots)$ where color i is associated with s_i. The packing coloring is the instance where $S = (1, 2, 3, \dots)$. The algorithm is for the decision problem when S is bounded by a constant t which is not the case for the packing coloring. Nevertheless, the packing chromatic number can be bounded for tress as stated by Theorem 10 from [10].

Theorem 10. *For any tree T with n vertices, $\chi_P(T) \leq \dfrac{n+7}{4}$,except when $n = 4$ or $n = 8$ in which case the bound is $\dfrac{n}{4} + 2$.*

3.2.5 Treewidth

Definition 20. ***A tree decomposition*** *of a graph $G = (V, E)$ is a tree T and a set $\{T_v\} \subseteq T$ of subtrees for each $v \in V$ such that $\forall v_1, v_2 \in V$ $v_1v_2 \in E \implies T_{v_1} \cap T_{v_2} \neq \emptyset$.*

Definition 21. *The* ***width*** *of a tree decomposition T of G is the maximum over all nodes $x \in V(T)$ of the number of subtrees in which x is in minus 1. More formally,*

$$\max_{x \in V(T)} \mid \{v \in V(G) : x \in V(T_v)\} \mid -1$$

Definition 22. *The* ***treewidth*** *of a graph G, denoted $tw(G)$, is the minimum width over all tree decomposition of G.*

This notion can be seen as an evaluation of how close a graph is from a tree. Several NP-hard problems like finding a maximum independent set , can be solved in polynomial time for type of graphs with bounded treewidth. For example, trees are the only graphs with treewidth 1. The (S-)packing coloring problem is one of these NP-hard problems. Some special cases listed below can be solved with a polynomial time algorithm.

1. The packing coloring when k is fixed proved by [8].
2. The packing coloring when the diameter is also bounded proved by [8].
3. The S-packing coloring when S is bounded by a constant stated in [8] based on [3].

3.2.6 Cartesian product

This section considers Cartesian product of two graphs. The Cartesian product of two sets A and B denoted $A \times B$ is the set $C = \{(a, b) | a \in A, b \in B, \}$. We can observe that $|C| = |A||B|$. The Cartesian product of 2 graphs is based on it and is defined as follow :

Definition 23. *The* ***Cartesian product*** *of two graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$ is a graph with $V_1 \times V_2$ as set of vertices and there is an edge between (v_1, v_2) and $(v_1\prime, v_2\prime)$ with $v_1, v_1\prime \in V_1$ and $v_2, v_2\prime \in V_2$ iff $v_1v_1\prime \in E_1$ and $v_2 = v_2\prime$ or $v_2v_2\prime \in E_2$ and $v_1 = v_1\prime$.*

Some results about upper and lower bounds of χ_P are stated in [4] and presented in this section via Theorem 11 and corollary 2.

Theorem 11. *Let G be a connected graphs with $n \geq 2$. Then*

$\chi_P(G \times H) \geq (\chi_P(G) + 1)|H| - diam(G \times H)(|H| - 1) - 1$ *with diam the diameter*

$\chi_P(G \times H) \geq (\chi_P(H) + 1)|G| - diam(G \times H)(|G| - 1) - 1$ *by commutativity of the Cartesian product*

Corollary 2. *Let $n \geq 2$. Then for any graph G*

$$\chi_P(G \times K_n) \geq n\chi_P(G) - (n - 1)diam(G)$$

Proof. From Theorem 11 we know :

$$\chi_P(G \times H) \geq (\chi_P(G) + 1)|H| - diam(G \times H)(|H| - 1) - 1$$

$$\chi_P(G \times K_n) \geq (\chi_P(G) + 1)|K_n| - diam(G \times K_n)(|K_n| - 1) - 1$$

$$\chi_P(G \times K_n) \geq (\chi_P(G) + 1)n - diam(G \times K_n)(n - 1) - 1 \text{ by definition of } K_n$$

$$\chi_P(G \times K_n) \geq (\chi_P(G) + 1)n - (diam(G) + 1)(n - 1) - 1 \text{ by } diam(G \times K_n) = diam(G) + 1$$

$$\chi_P(G \times K_n) \geq n\chi_P(G) - (n - 1)diam(G)$$

□

Definition 24. *The* ***2-independence number*** *of a graph is the size of the largest union of two independent set of G.*

[4] also found an upper bound for the Cartesian product of any graph G and H :

$$\chi_P(G \times H) \leq |G||H| - \alpha(G)\alpha(H) - min\{|G| - \alpha(G), |H| - \alpha(H)\} + 1$$

If H is bipartite then the upper bound can be reduced to :

$$\chi_P(G \times H) \leq |G||H| - \frac{\alpha_2(H)}{2} + 1 \text{ with } \alpha_2 \text{ the 2-independence number}$$

They also left another upper bound as open question :

$$\chi_P(G \times H) \leq max\{\chi_P(G)|H|, \chi_P(H)|G|\}$$

3.2.7 Lattices graph

Definition 25. *A **lattice graph** $L_{m,n}$ is the graph $K_m \times K_n \forall n, m \in \mathbb{Z}$.*

The packing chromatic number of K_n is n because it has a clique of size n so n is both an upper and lower bound. We can apply corollary 2 , to find a lower bound of $lb = nm - (n-1) = n(m-1) - 1$. By the results from the previous section, an upper bound is $nm - 1 - n = n(m-1) - 1$ if $n < m$. We can conclude that $\chi_P(L_{m,n}) = n(m-1) - 1$ for all lattice graph.

There is also special lattice like infinite triangular or hexagonal lattice. [4] proved that an hexagonal lattices has a packing chromatic number between 7 and 8 by inter alia illustrating a coloring with an eight coloring pattern illustrated in Figure 3.4. However, [9] showed that this number is unbounded for infinite triangular lattices

The infinite 2D square lattice also called grids. [10] studied this types of graph and found patterns for special types of grid. Its main result is a pattern that uses at most 23 colors for any grids. This bounds is decreased by Schwenk as said in [10] by proving that a packing coloring of the infinite grid uses at most 22 colors.

Definition 26. *An **hypercube** is a graph denoted Q_n where each vertex represents one of the possible vector $0, 1^n$ thus having 2^n vertices. There is an edge between 2 vertex iff the Hamming distance between their vectors is exactly one.*

In higher dimension, this number is unbounded even for the 3D infinite square grid as stated in [9]. For example, the packing chromatic number of hypercubes is approximated by $(\frac{1}{2} - \mathcal{O}(\frac{1}{n}))2^n$ in [10]. Nonetheless hypercubes are still studied in [16]. They found bounds or exact value for hypercubes with a small n. Their results and those already found in [10] are summarized in Table 3.1. The exact value was known for $n \leq 5$ but now it is extended to $n \leq 8$. Moreover, the gap between lower and upper bound is shrink for higher n.

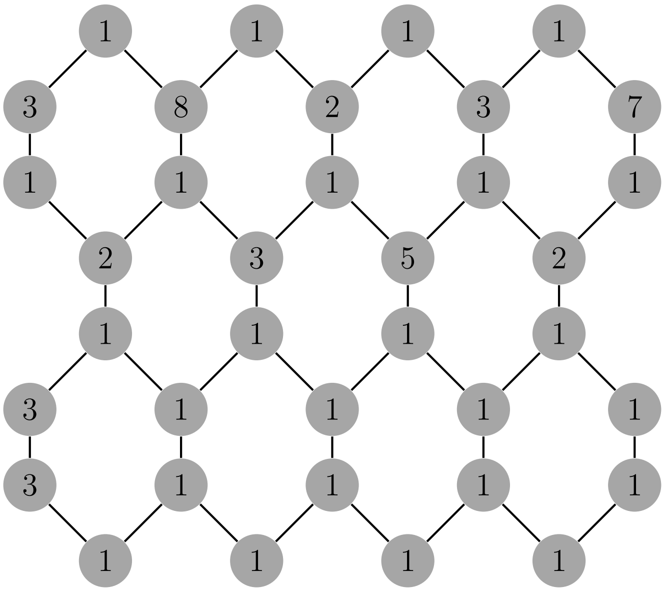

Figure 3.4: Template for a 8 coloring of a hexagonal lattice

n	1	2	3	4	5	6	7	8	9	10	11
Lower bound	2	3	5	7	15	25	49	95	198	395	794
Upper bound	2	3	5	7	15	25	49	95	211	421	881
Gap	0	0	0	0	0	0	0	0	13	26	87

Table 3.1: Bounds for $\chi_P(Q_n)$

3.2.8 Subdivisions

Definition 27. *A **subdivision** of a graph $G = (V,E)$ is a graph H where one or several edge of G are replaced by a path between the two endpoints.*

Definition 28. *A **subdivision graph S(G)** of a graph $G = (V,E)$ is a graph H where all edge of E are subdivided by a path of one vertex thus $V(H) = V \cup E(G)$.*

Figure 3.5 illustrates a graph and its subdivision graph with their packing coloring. In [4], they proved the general Theorem 12 and the more specific lemma 1.

Theorem 12. *For any connected graph G with at least three vertices.*

$$\omega(G) + 1 \leq \chi_P(S(G)) \leq \chi_P(G) + 1$$

Lemma 1. *For any $n \geq 3$, $\chi_P(S(K_n)) = n + 1$*

We can compare lemma 1 with result obtained in previous section that $\chi_P(K_n) = n$ so subdividing a complete graph only increase its χ_P by one. Moreover [10] proved that $\chi_P(S(G)) = 3$ for any connected bipartite graph with at least 2 edges.

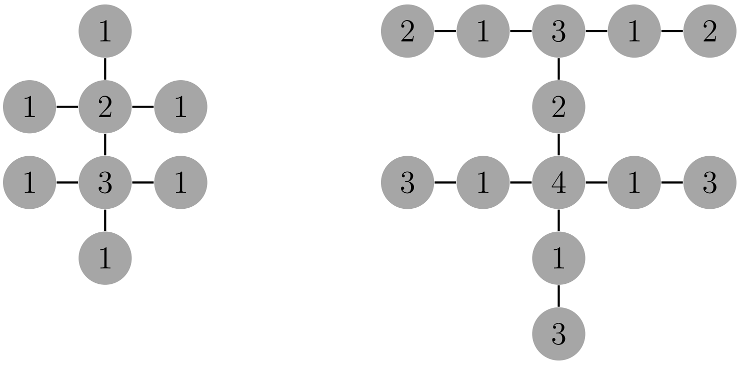

Figure 3.5: Optimal packing coloring of a graph and its subdivision graph

3.3 Formulations

This section is about one formulation made by [13] which is an adaptation of the one presented in Section 2.4. They also were able to do find bounds for certain type of graphs.

The formulation also includes a variable $x_{ic} \forall i \in V \forall c \in C$ which have the same meaning as before(Recall $x_{vc} = 1$ iff vertex v has color i). This formulation uses another parameter d_{ij} which is the distance between two vertices because knowing the edges is not enough anymore.

$$
\begin{aligned}
&\mathit{minimize} && z && && (3.2)\\
&\mathit{subject\ to} && \sum_{i=0}^{k} x_{vi} = 1 && \forall v \in V && (3.3)\\
& && x_{ui} + x_{vi} \leq 1 && \forall u, v \in V \mid d_{uv} \leq i,\ \forall i \in \{1, \dots, k\} && (3.4)\\
& && i x_{vi} \leq z && \forall v \in V\, \forall i \in \{1, \dots, k\} && (3.5)\\
& && x_{vi} \in \mathbb{B} && \forall v \in V,\ \forall i \in \{1, \dots, k\} && (3.6)\\
& && z \in \mathbb{Z} && && (3.7)
\end{aligned}
$$

The variable z to be minimized in 3.2 represents the number of color used. A vertex can only have one color as stated by constraint 3.3. The constraint 3.4 applies the constraint on the minimum distance between two vertices colored with the same color. Finally, 3.5 gives meaning to z because it restricts it to be at least the greatest color used but z is minimized thus z is well-defined. This formulations is very close to 2.4. This formulation has $1 + nk = \mathcal{O}(n^2)$ variables and $n + \mathcal{O}(n^2) + nk = \mathcal{O}(n^2)$. We can observe that a color c is used at most once if $c \geq \mathit{diam}(G)$ otherwise 2 vertices colored c would be separated by at most $\mathit{diam}(G)$ so the constraint 3.4 is violated. This simple observation allows to reduce the number of variables and constraints with this new formulations also from [13].

$$\begin{aligned}
&\textit{minimize} \quad & D-1+|V(G)|-\sum_{v\in V}\sum_{i=1}^{D-1} x_{vi} \quad & \text{D is diam(G)} && (3.8)\\
&\textit{subject to} \quad & \sum_{i=0}^{k} x_{vi} = 1 \quad & \forall v \in V && (3.9)\\
& & x_{ui}+x_{vi} \leq 1 \quad & \forall u,v \in V \mid d_{uv} \leq i, \forall i \in \{1,...,D-1\} && (3.10)\\
& & x_{vi} \in \mathbb{B} \quad & \forall v \in V,\ \forall i \in \{1,\ldots,D-1\} && (3.11)\\
& & & && (3.12)
\end{aligned}$$

The cost function 3.8 is adapted by adding $D-1$ to the number of vertices not colored by the first $D-1$ colors because every vertex not colored by them uses its own color. The constraints are only an adaptation of the previous formulation without 3.5 because z is no longer needed. This formulation has the problem that 3.8 represents $\chi_P(G)$ iff $\chi_P(G) >= D-1$ otherwise it can be deduced from the x variables. These formulations using only Boolean variables thus it can be transformed to a SAT formula, as done in this paper, to be able to use a specialized solver.

The number of variables does not depend on the number of color anymore because there is $n(D_1)$ variables. The number of constraints is still in $\mathcal{O}(n^2)$ but is reduced because there is not the $\mathcal{O}(n^2)$ constraint from 3.5 and 3.10 generates less constraints than 3.4.

Definition 29. *An* ***distance-k independence set*** *of a graph $G = (V,E)$ is a set of vertices $I_k \subseteq V$ s.t. $\forall u, v \in I_k d_{uv} > i$.*

Definition 30. *The* ***distance-k independence number*** *of a graph $G = (V,E)$ denoted $\alpha_k(G)$ is the maximum cardinality of a distance-k independence set of G.*

This paper generates the notion of independence set with distance-k independence set and also denotes by $\beta_i(G)$ the largest number of vertices that can be colored with i colors. By using variables from the formulation above, these values must respect these constraints :

$$\sum_{v\in V} x_{vi} \leq \alpha_i \forall i \in \{1,\ldots,k\} \tag{3.13}$$

$$\sum_{c=1}^{i}\sum_{v\in V} x_{vc} \leq \beta_i \forall i \in \{1,\ldots,k\} \tag{3.14}$$

Constraint 3.13 claims that $\alpha_i(G)$ is at least the number of vertices colored i because these vertices forms a distance-i independence set. Constraint 3.14 claims that $\beta(G)$ is at least the number of vertices colored i or less. The value of α_k can be computed using algorithms for maximum cliques. However, the computation of β_k is more difficult. Indeed, [13] was only able to compute β_2 and β_3 with linear programming.

Chapter 4

Independent set color formulation for packing coloring

4.1 Introduction

This chapter focuses on an adaptation of the independent set formulation for the PC problem. Firstly, Section 4.2 explains the idea and presents the new formulation. Then, Section 4.3 describes the implementation. Section 4.4 presents computational results and their interpretation. Finally, Section 4.5 presents a new way to solve the formulation using standard branch and price algorithm.

4.2 Formulation

4.2.1 Idea

As before, there is one variable per independent set. Nevertheless, a specific independent set cannot be used for any color. Therefore, the set of independent set I is divided in classes as represented in Figure 4.1. The class I_j is the set of independent sets that can be used for colors $1, 2, \ldots, j$ and not colors $> j$. The ∞ class can be used for any color due to the lack of path between its vertices. There is D classes with D the diameter because the distance between 2 connected vertices cannot exceed D thus I_j with $j \geq D$ is empty. For each class, we limit ourselves to the maximal sets.

Definition 31. *A set of vertices S belonging to class I_j is* ***maximal*** $\iff \forall v \in V, S \cup v \notin I_j$.

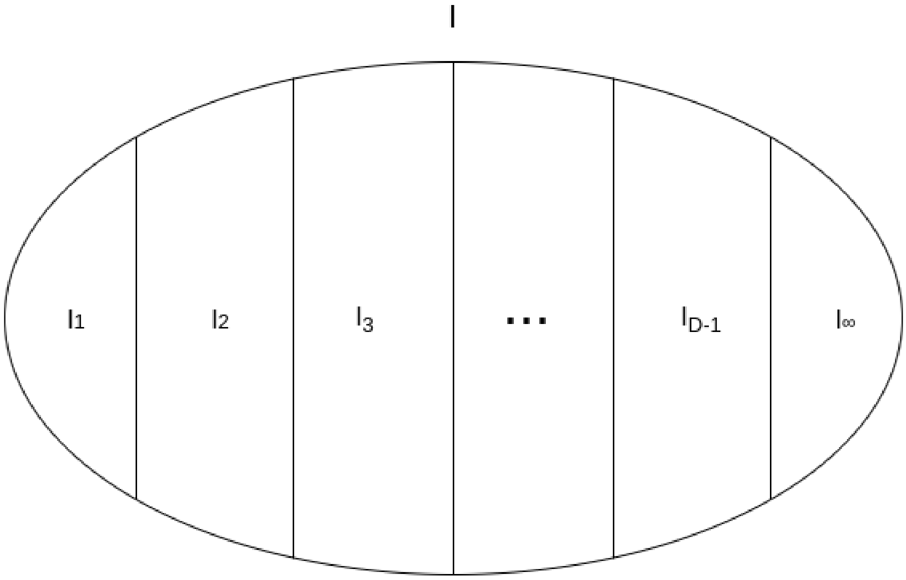

Figure 4.1: Class separation of I

4.2.2 Formulation

The formulation denoted IPC need several variables and parameters which are the following :

$$\forall i \in I \ x_i = \begin{cases} 1 & \text{if the independent set } i \text{ is taken to represent a color} \\ 0 & \text{otherwise} \end{cases}$$

As said, there is one variable by set. But some parameters are needed to complete the formulation. First, we need to know in which class are each set which was not needed for IC due to the non differentiation of colors. The classification is done in pre-processing as explained later in 4.3. Moreover, the content of each set as to be known like in IC. These are given by parameters denoted respectively d and c.

The class of a independent set i is d_i.

$$\forall i \in I, \forall v \in V \ c_{iv} = \begin{cases} 1 & \text{if the vertex v is in the independent set } i \\ 0 & \text{otherwise} \end{cases}$$

All parameters and variables are defined so we can present the formulation :

$$\begin{aligned} &\mathit{minimize} && \sum_{i\in I} x_i && (4.1)\\ &\mathit{subject\ to} && \sum_{i\in I|v\in i} x_i = 1 \quad \forall v \in V && (4.2)\\ & && \sum_{i\in I|d_i\leq k} x_i \leq k \quad \forall k \in [1; D-1] && (4.3)\\ & && x_i \in \mathbb{B} \qquad \forall i \in I && (4.4) \end{aligned}$$

The objective function 4.1 minimizes the number of independent set chosen thus the number of color. Each vertex is in one and only one chosen independent set so has a clearly defined color is fixed by Constraint 4.2. Note that these two constraints are identical to the classical version of this formulation. Constraint 4.3 represents the fact that we cannot take too much sets from each partition. Indeed, we can take at most j element from I_j because they can be used only for color 1 to j. We must also take into account the number of element taken from class lower than I_j. For example, if two elements from I_2 have already been taken in a partial solution then at most two elements of I_4 could be in the complete solution because colors 1 and 2 are already represented by I_2 thus I_4 can only take color 3 and 4.

Compared to IC, IPC has the same number of variable but the number of constraints increases from N to $N + D$ which is still in $\mathbb{O}(N)$.

4.3 Implementation

IPC needs some pre-processing including finding all independent set and classifying them. All independent set are found by a classic backtracking generating all binary string of size N representing which vertex is taken in the set. A branch can be pruned if we add a vertex sharing an edge with one of the vertices from the current independent set which make it loses its independent property by definition. Another case is when an independent set is not maximal. It is the case where we can add a vertex by keeping an independent set with the same Class. Each solution and its class is stored on a binary file to save space. This file is used as input for the "optimization" part/module of the code. This separation is used to make each part independent in term of code and memory usage. The optimization part is implemented with cplex library in c++. The choose of cplex over other possibilities like xpress or NEOS server is based on the price, size of instances[1], compatibility with OS and integration in programming language. The pre-processing part is in $\mathbb{O}(2^N + N^2) = \mathbb{O}(2^N)$ due to exponential complexity of backtracking. In practice it can be much less due to pruning's efficiency being inversely proportional to the density of the graph. The algorithm used to solve a formulation is simply using the method solve from Cplex as black box.

[1] limitatioçn in free version

4.4 Result

The algorithm was tested on instances from DIMACS for graph coloring encoded in .col format found http://mat.gsia.cmu.edu/COLOR/instances.html. We tested two formulations : the presented in [13] and the one presented in Section 4.2. The hardware and software configuration on which the tests were conducted has as characteristic :

- OS : Linux mint 18.1 (kernel 4.4.0).
- CPU : Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz with 4 cores (8 threads).
- RAM : 32GB at 2400 MHz + 4GB of swap partition.
- Secondary memory : SSD 120GB
- Cplex 12.7 with default parameters.

The results are represented on Table 4.1. The preprocessing time for VC is not indicated because it is only a parser and take less than one ms which is negligible in comparison of the total time.

Name	N	density	time VC′ (s)	Pre-Time IS (s)	Time IS (s)	# color
DSJC125.5	125	0.5	0.965143	0.930401	0.213191	116
DSJC125.9	125	0.9	0.946579	0.008601	0.003331	122
DSJC250.5	250	0.5	74.786779	102.042875	9.934149	239
DSJC250.9	250	0.9	10.150312	0.094591	0.020786	246
DSJC500.9	500	0.9	45.448983	1.579720	0.074801	496
DSJC1000.9	1000	0.9	735.636395	38.864877	1.069712	995
flat300 20	300	0,2375	162.278250	796.447398	140.888260	286
flat300 26	300	0.24	319.709602	620.181133	54.438221	289
flat300 28	300	0.241	260.156268	602.066899	58.223397	289
Latin square 10	900	0,38	39.507431	523.119600	0.347178	891

Table 4.1: Computation time per instances for two formulations

We can observe that it takes much longer time for preprocessing than solving for the IS formulation. Preprocessing time changes proportionally with N and inversely proportionally with density. On one hand, there is generally more independent set/subset when N increases with same density. On the other hand, a higher density means more edges thus less independent set. The density has a bigger impact than N on the time. For instance, going from 250 to 500 vertices with a density of 0.9 multiplies only the time by 3 but keeping 250 vertices and going from 0.9 to 0.5 multiply the time by nearly 1000 times. For VC formulations, the most important factor is N because there is N^2 variables. A higher density decreases the time but less than for IS. It can be explained by the lower number of constraint because there are more adjacent vertices which only need to not both have the color one. The density has less impact due to the fact that the number of variables is constant and the reduction in cardinality of constraint is nothing compared to the reduction of variables in IS.

The preprocessing time increases exponentially with N but the cplex time is lower than the $VC\prime$ formulation

4.5 Branch and Price

4.5.1 Introduction

A big percentage of computation time is used on preprocessing and the solver time is lower than $VC\prime$ so can we reduce time by using the same formulation ? Indeed, it is the case, a way to reduce it is by using the column generation method. This general algorithm is often uses when dealing with problem having a large number of variables.

4.5.2 Algorithm

The algorithm follows this structure :

1. Solve linear relaxation master problem with restricted columns.
2. Get optimal dual value π.
3. Solve subproblem with coefficient π.
4. Add Column from subproblem to main problem.
5. Repeat until subproblem solution ≤ 0.

This template introduces new concepts. Firstly, the idea of adding column which represents variable. The idea is to add a variable at each turn until the solution is optimal which means here that considering new independent sets will not improve the solution (coloring) quality. If we add column then all columns are not included initially. Indeed, we talk about **restricted** master problem. We start with a initial set of columns from which a solution exists. The master problem is the problem to solve. Generally; it has a variable for each solution from the initial problem. For instance, there can be a variable for each subset of location for the facility location problem which include a exponential number of variable which needs a lot of resource (memory, time,...) to be solved. In our case, we have to select several variables to have a solution but te number of variables is still exponential to N.

We solve the **linear relaxation** which means that each variable is considered as a real value instead of integer/boolean ones in order to have the dual variable coefficient π and be able to compute reduced costs of the missing columns. Nonetheless, it would be inefficient to computed the reduced cost for all missing columns because we would need to enumerate all columns so all independent set which is equivalent to the preprocessing which we want to eliminate. As result, the choice of the new column is expressed as a MIP[2] called subproblem which tends to minimize the reduced cost. This problem can also have constraints to ensure integrity of the new column. In our problem the new column must be an independent set.

4.5.3 Implementation

The section presents the different elements of branch and price adapted to our problem and formulation IS. First, we describe the subproblem with the dual of linear relaxation of the master problem.

Master problem • The matrix d description :

$$\forall i \in I, \forall k \in [1;D]\ d_{ik} = \begin{cases} 1 & \text{if the independent set } i \text{ is in } \bigcup_{j=1}^{k} I_j \\ 0 & \text{otherwise} \end{cases}$$

- I_R is the restricted set of columns.

$$\begin{aligned} & \mathit{minimize} \sum_{i \in I} x_i && && (4.5) \\ & \mathit{subject\ to} \sum_{i \in I} c_{iv} x_i = 1 && \forall v \in V\ (\pi_v) && (4.6) \\ & \sum_{i \in I_k} d_{ik} x_i \leq k && \forall k \in [1; D-1]\ (\alpha_k) && (4.7) \\ & x_i \in \mathbb{R} && \forall i \in I_R && (4.8) \end{aligned}$$

Dual

$$\begin{aligned} & \mathit{maximize} \sum_{v \in V} \pi_v + \sum_{k=1}^{D-1} k \alpha_k && && (4.9) \\ & \mathit{subject\ to} \sum_{v \in V} c_{iv}\ \pi_v + \sum_{k=1}^{D-1} d_{ik} \alpha_k \leq 1 && \forall i \in I_R && (4.10) \\ & \pi_v \in \mathbb{R} && \forall v \in V && (4.11) \\ & \alpha_k \in \mathbb{R} && \forall k \in [1; D-1] && (4.12) \\ & && && (4.13) \end{aligned}$$

This gives us a reduced cost to minimize :

$$1 - \sum_{v \in V} c_{iv}\ \overline{\pi_v} + \sum_{k=1}^{D-1} d_{ik} \overline{\alpha_k} \text{ with } \overline{x} \text{ means the optimal value of x}$$

[2] Mixed Integer Problem

Subproblem

$$
\begin{aligned}
&\mathit{maximize} \sum_{v \in V} y_v\ \overline{\pi_v} + \sum_{k=1}^{D-1} z_k \overline{\alpha_k} && && (4.14)\\
&\mathit{subject\ to}\ z_k \geq z_{k+1} && \forall k \in [1; D-2] && (4.15)\\
&x_{uvk} \leq y_u && \forall u, v \in V^2, \forall k \in [1; D-1] && (4.16)\\
&x_{uvk} \leq z_k && \forall u, v \in V^2, \forall k \in [1; D-1] && (4.17)\\
&y_u + y_v + z_k - 2 \leq x_{uvk} && \forall u, v \in V^2, \forall k \in [1; D-1] && (4.18)\\
&x_{uvk}(k+1) \leq d_{uv} x_{uvk} && \forall u, v \in V^2, \forall k \in [1; D-1] && (4.19)\\
&z_1 = 1 && && (4.20)\\
&y_v \in \mathbb{B} && \forall v \in V && (4.21)\\
&z_j \in \mathbb{B} && \forall j \in [1; D-1] && (4.22)\\
& && && (4.23)
\end{aligned}
$$

$$\forall v \in V\ y_v = \begin{cases} 1 & \text{if the vertex v is in the new column} \\ 0 & \text{otherwise} \end{cases}$$

$$\forall k \in [1; D]\ z_k = \begin{cases} 1 & \text{if the column is in } \bigcup_{j=1}^{k} I_j \\ 0 & \text{otherwise} \end{cases}$$

Finally, x_{uvk} represents the SAT formula $y_u \wedge y_v \wedge z_k$. This variable meaning is enforced by Constraints 4.16 - 4.18. Constraint 4.15 ensures that an independent set cannot be in $\bigcup_{j=1}^{k-1} I_j$ but not in $\bigcup_{j=1}^{k} I_j$ since the first one includes the second. If two vertices are taken and the independent set can be used for color $k+1 \iff z_k = 1$ then the distance between u and v is at least $k+1$. This condition is defined by 4.19. Finally, Constraint 4.20 ensures that we have an independent set.

Bibliography

[1] K. Appel and W. Haken. Every planar map is four colorable. part i: Discharging. *Illinois J. Math.*, 21(3):429–490, 09 1977.

[2] Laura Bahiense, Yuri Frota, Thiago F. Noronha, and Celso C. Ribeiro. A branch-and-cut algorithm for the equitable coloring problem using a formulation by representatives. *Discrete Applied Mathematics*, 164, Part 1:34 – 46, 2014. Combinatorial Optimization.

[3] R. B. Borie. Generation of polynomial-time algorithms for some optimization problems on tree-decomposable graphs. *Algorithmica*, 14(2):123–137.

[4] Boštjan Brešar, Sandi Klavžar, and Douglas F. Rall. On the packing chromatic number of cartesian products, hexagonal lattice, and trees. *Discrete Applied Mathematics*, 155(17):2303 – 2311, 2007.

[5] Manoel Campêlo, Victor A. Campos, and Ricardo C. Corrêa. On the asymmetric representatives formulation for the vertex coloring problem. *Discrete Applied Mathematics*, 156(7):1097 – 1111, 2008. {GRACO} 20052nd Brazilian Symposium on Graphs, Algorithms and Combinatorics.

[6] Manoel Campêlo, Ricardo Corrêa, and Yuri Frota. Cliques, holes and the vertex coloring polytope. *Information Processing Letters*, 89(4):159 – 164, 2004.

[7] Jan Ekstein, Přemysl Holub, and Olivier Togni. The packing coloring of distance graphs. *Discrete Applied Mathematics*, 167:100 – 106, 2014.

[8] Jiří Fiala and Petr A. Golovach. Complexity of the packing coloring problem for trees. *Discrete Applied Mathematics*, 158(7):771 – 778, 2010. Third Workshop on Graph Classes, Optimization, and Width Parameters Eugene, Oregon, USA, October 2007.

[9] Arthur S. Finbow and Douglas F. Rall. On the packing chromatic number of some lattices. *Discrete Applied Mathematics*, 158(12):1224 – 1228, 2010. Traces from LAGOS'07 {IV} Latin American Algorithms, Graphs, and Optimization Symposium Puerto Varas - 2007.

[10] Wayne Goddard, Sandra M. Hedetniemi, Stephen T. Hedetniemi, John M. Harris, and Douglas F. Rall. Broadcast chromatic numbers of graphs. In *16TH CUMBERLAND CONFERENCE ON COMBINATORICS, GRAPH THEORY, AND COMPUTING , 2003*, 2003.

[11] Anuj Mehrotra and Michael A. Trick. *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, chapter A Branch-And-Price Approach for Graph Multi-Coloring, pages 15–29. Springer US, Boston, MA, 2007.

[12] Neil Robertson Daniel P. Sanders Paul Seymour and Robin Thomas. A new proof of the four-colour theorem. *electronic research announcements of the american mathematical society*, 2(1):17–25, 8 1996.

[13] Zehui Shao and Aleksander Vesel. Modeling the packing coloring problem of graphs. *Applied Mathematical Modelling*, 39(13):3588 – 3595, 2015.

[14] Michael Sipser. *Introduction to the theory of computation*. Course Technology, Inc, thomson edition, 2006.

[15] Christian Sloper. An eccentric coloring of trees. *Australasian Journal of Combinatorics*, 29:309 – 321, 2004.

[16] Pablo Torres and Mario Valencia-Pabon. The packing chromatic number of hypercubes. *Discrete Applied Mathematics*, 190–191:127 – 140, 2015.