



UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES
DEPARTEMENT DES MATHÉMATIQUES



Master Mathématique et Application au Calcul Scientifique (MACS)

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du Diplôme de Master Sciences et Techniques
(MST)

Coloration des graphes : structures et algorithmes

Réalisé par: JETTIOUI Khadija

Encadré par: Pr. HILALI Abdelmajid

Soutenu le 15 juin 2017

Devant le jury composé de:

- | | |
|-----------------------------|-------------------------------------------|
| - Pr. HILALI Abdelmajid | Faculté des sciences et Techniques de Fès |
| - Pr. ELHILALI ALAOUI Ahmed | Faculté des sciences et Techniques de Fès |
| - Pr. EL KHOMSSI Mohammed | Faculté des sciences et Techniques de Fès |
| - Pr. MESSAOUD El Hassania | Université privée de Fès |

Année Universitaire 2016 / 2017

FACULTE DES SCIENCES ET TECHNIQUES FES – SAISS

☒ B.P. 2202 – Route d'Imouzer – FES

☎ 212 (0)5 35 61 16 86 – Fax: 212 (0)5 35 60 82 14

Remerciements

Ce mémoire n'aurait jamais vu le jour sans le soutien, la générosité et les encouragements continus de mon encadrant **Pr. HILALI Abdelmajid**, je voudrais le remercier chaleureusement.

Mes respectueux remerciements s'adressent aussi à tous les membres du jury **Pr. ELHILALI ALAOUI Ahmed**, **Pr. EL KHOMSSI Mohammed** et **Pr. MESSAOUD El Hassania** qui m'ont fait l'honneur de bien vouloir juger ce travail et de l'enrichir par leurs remarques et critiques.

Je tiens à remercier une deuxième fois tout particulièrement et à témoigner toute ma reconnaissance et ma gratitude à mon Professeur **ELHILALI ALAOUI Ahmed**, responsable de Master **MACS** pour l'extrême richesse de son enseignement et son soutien durant ces deux années.

Egalement, je tiens à remercier le corps professoral de ma formation à la Faculté des Sciences et Techniques de Fès pour leur contribution à ma formation et pour leurs enseignements précieux et multidisciplinaires.

Dédicace

Je dédie ce modeste travail

A mes chers parents ma mère et mon père

Pour leur patience, leur amour, leur soutien et leurs encouragements.

A mes frères et soeurs.

A toute ma famille.

A tous (toutes) mes ami(e)s.

A tous ceux qui ont contribué de près ou de loin à l'édition de ce rapport.

Table des matières

Introduction	8
1 Notions de base	11
1.1 Introduction	11
1.2 Les graphes	11
1.2.1 Les concepts de base	11
1.2.2 Chaînes, cycles, connexité, trous et antitrous	12
1.2.3 Quelques graphes particuliers	13
1.3 Les algorithmes	18
1.3.1 Problèmes, algorithmes, complexité	19
1.3.2 Les problèmes P , NP , NP -complets et NP -difficiles	19
1.4 Conclusion	20
2 Coloration des graphes	21
2.1 Introduction	21
2.2 Un peu d'histoire	21
2.3 Domaines d'applications	22
2.4 Coloration des sommets	22
2.4.1 Définitions et propriétés	22

2.4.2	Complexité	28
2.4.3	Coloration des graphes planaires	28
2.4.4	Coloration des graphe parfaits	28
2.5	Coloration des arêtes	30
2.5.1	Définitions et propriétés	30
2.5.2	Complexité	34
2.6	Conclusion	34
3	Quelques algorithmes de coloration	35
3.1	Introduction	35
3.2	Graphes quelconques : des algorithmes d'approximation	35
3.2.1	Coloration séquentielle : l'algorithme glouton	35
3.2.2	Une première heuristique, Welsh & Powell	37
3.2.3	Deuxième heuristique, DSATUR	39
3.3	Graphes triangulés : un algorithme linéaire de résolution exacte	44
3.4	Conclusion	49
4	Applications et résultats expérimentaux	50
4.1	Introduction	50
4.2	Allocation des fréquences (Gestion du spectre de fréquence et implémenta- tion des réseaux de télécommunications)	50
4.2.1	Introduction	50
4.2.2	Quelques définitions	51
4.2.3	Les objectifs de la gestion du spectre	52
4.2.4	Les interférences	52
4.2.5	Problématique	53
4.2.6	Modélisation mathématique du problème par la théorie des graphes	53
4.2.7	Problèmes tests	54

4.2.8	Résultats expérimentaux	56
4.2.9	Comparaison des résultats	60
4.3	Conclusion	61
Conclusion et perspectives		62
Bibliographie		63

Table des figures

1.1	Exemple d'un graphe biparti	13
1.2	(-a-) un graphe triangulé (-b-) un graphe qui n'est pas triangulé car il contient un cycle de longueur 4 qui n'a pas de corde.	14
1.3	Exemple d'un graphe planaire	17
1.4	Les faces d'un graphe planaire	18
2.1	Exemple de coloration d'arêtes d'un graphe	31
4.1	Un réseau d'antennes	51
4.2	Graphe d'interférences correspondant au tableau ci-dessus, dont les som- mets sont les stations, et les arêtes relient deux stations qui interfèrent entre elles	56
4.3	Coloration obtenue par l'algorithme de coloration séquentielle	57
4.4	Coloration obtenue par l'algorithme de Welsh and Powell	58
4.5	Coloration obtenue par l'algorithme de DSATUR	59
4.6	Coloration obtenue par l'algorithme de LexBFS	60

Liste des tableaux

3.1	Tableau de coloration de graphe par welsh and powell	39
3.2	Tableau de Parcours en Largeur Lexicographique	49
4.1	Tableau des distances entre les stations	55

Introduction

La théorie des graphes constitue une branche à part entière des mathématiques, grâce aux travaux de Cayley, Menger, König, Berge, Erdős, etc. Son histoire débute avec les travaux d'Euler au *XVIII^e* siècle et trouve son origine dans l'étude de certains problèmes, tels que celui des ponts de Königsberg (les habitants de Königsberg se demandaient s'il était possible, en partant d'un quartier quelconque de la ville, de traverser tous les ponts sans passer deux fois par le même et de revenir à leur point de départ), la marche du cavalier sur l'échiquier ou le problème de coloriage de cartes.

La théorie des graphes s'est alors développée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales,...

De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques,...

Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et d'arcs. Les derniers travaux en théorie des graphes sont souvent effectués par des informaticiens, du fait de l'importance qu'y revêt l'aspect algorithmique.

Dans ce mémoire, on s'intéresse à la coloration des graphes qui est un problème ancien, comme en témoigne le fameux problème des 4 couleurs posé par Francis Guthrie en 1852 : est-il possible de colorier toute carte géographique avec au plus 4 couleurs de sorte que 2 régions qui ont une frontière en commun aient des couleurs différentes ?

Depuis, c'est un champ majeur et très actif de la théorie des graphes avec un nombre important de conjectures, comme celle de Hedetniemi à propos du nombre chromatique du produit direct de deux graphes ou celle de Reed qui lie le nombre chromatique, la taille maximum des cliques et le degré maximum d'un graphe.

La coloration de graphes permet de modéliser de nombreux problèmes réels, depuis le pla-

cement de personnes autour d'une table ou de pièces sur un échiquier jusqu'aux différents problèmes d'ordonnancement et de planning de la vie de tous les jours (réservation de ressources, logistique, ...) et notamment dans le domaine des réseaux/télécom.

Comme nous allons le voir, il n'y a pas une seule façon de colorier les graphes mais plusieurs (un très grand nombre même). On peut déjà vouloir colorier différents éléments d'un graphe (les sommets, les arêtes, les faces, un mélange de ces éléments, des sous-structures, ...) et à ceci peut s'ajouter différentes contraintes.

La contrainte la plus courante est celle de la propriété : deux éléments voisins doivent avoir des couleurs différentes. D'ailleurs, sauf mention contraire, toutes les colorations considérées dans ce document seront propres, même si ce n'est pas toujours précisé.

Bien que différents éléments d'un graphe puissent être coloriés, on peut toujours se ramener à un problème de coloration de sommets. C'est donc le paramètre de coloration le plus général (et le plus étudié).

Voici un résumé des différents chapitres qui composent ce travail.

Chapitre 1 : Notions de base

Dans ce premier chapitre, nous rappelons toutes les définitions de la théorie des graphes nécessaires à une bonne compréhension de la suite de ce mémoire, et certaines classes de graphes que nous rencontrerons souvent dans les chapitres suivants. Nous donnons également un bref aperçu de la théorie de la complexité des algorithmes en insistant sur les notions fondamentales de classes de problèmes P et NP .

Chapitre 2 : Coloration des graphes

Ce chapitre traite le problème de la coloration. Dans la première partie de ce chapitre, on s'intéresse à la coloration des sommets, nous commençons par donner quelques définitions et propriétés, ensuite, nous présentons quelques résultats concernant la coloration de quelques graphes particulières. La deuxième partie est consacrée à la coloration des arêtes et ses propriétés. Mais avant ces deux parties, nous parlons un peu de l'histoire et les domaines d'applications de la coloration des graphes.

Chapitre 3 : Quelques algorithmes de coloration

Dans ce chapitre, nous présentons quelques algorithmes permettant de colorier les graphes, ils seront utilisés par la suite dans le chapitre des applications. Chaque algorithme est illustré par un exemple.

Chapitre 4 : Applications et résultats expérimentaux

Dans ce dernier chapitre, nous abordons la partie pratique : l'implémentation en C des différents algorithmes présentés dans le chapitre 3, quelques applications de la coloration des graphes dans le monde réel, ainsi qu'une étude comparative des résultats expérimentaux obtenus avec les divers algorithmes appliqués à ces applications réelles.

NOTIONS DE BASE

1.1 Introduction

Dans ce premier chapitre, on va rappeler toutes les définitions de la théorie des graphes nécessaires à une bonne compréhension de la suite de ce mémoire, et certaines classes de graphes que nous rencontrerons souvent dans les chapitres suivants . Nous donnons également un bref aperçu de la théorie de la complexité des algorithmes en insistant sur les notions fondamentales de classes de problèmes P et NP .

1.2 Les graphes

1.2.1 Les concepts de base

Si V est un ensemble fini quelconque, alors $[V]^2$ désigne l'ensemble des parties de V qui ont exactement 2 éléments. Un graphe simple G est un couple d'ensembles (V, E) tel que $E \subseteq [V]^2$. Les éléments de V sont appelés les sommets de G , et les éléments de E sont appelés les arêtes de G . Afin d'alléger les notations, la paire contenant u et v est notée uv , au lieu de $\{u, v\}$. Si G est un graphe, alors $V(G)$ désigne l'ensemble de ses sommets et $E(G)$ l'ensemble de ses arêtes.

Soient G un graphe simple et u, v deux sommets de G . Si $uv \in E(G)$ alors u et v sont dits adjacents ou voisins ; il est aussi dit que u voit v . Si $uv \notin E(G)$, alors u et v sont dits non-adjacents ou non-voisins ; il est aussi dit que u manque v .

Soit G un graphe simple, si $v \in V(G)$, alors l'ensemble des voisins de v est noté $N_G(v)$.

Remarquons que $v \notin N_G(v)$. Le degré d'un sommet v est défini par $d_G(v) = |N_G(v)|$.

Soit G un graphe simple, le graphe complémentaire de G noté \overline{G} , est défini par $V(\overline{G}) = V(G)$ et $E(\overline{G}) = \{uv; u \in V(G), v \in V(G) \text{ et } uv \notin E(G)\}$.

Soient $G = (V, E)$ un graphe simple et W un ensemble de sommets de G , un sous-graphe est un graphe contenu dans un autre graphe, c-à-d un graphe $H = (V_H, E_H)$ est un sous-graphe de $G = (V, E)$ si $V_H \subset V$ et $E_H \subset E$. Le sous-graphe de G induit par W est noté $G[W]$, c'est le graphe dont l'ensemble des sommets est W et dont l'ensemble des arêtes est $[W]^2 \cap E(G)$.

Un ensemble de sommets $X \subset V(G)$ d'un graphe G forme une clique si $\forall x, y \in X, xy \in E(G)$, une clique X est maximale si $\forall y \in V \setminus X, X \cup \{y\}$ n'est pas une clique, une clique x est maximum si toute clique Y satisfait $|X| \geq |Y|$, la taille d'une clique est son nombre de sommets, la taille d'une clique maximum de G est notée $\omega(G)$.

Un triangle est une clique de taille trois.

Un ensemble de sommets $X \subset V(G)$ d'un graphe G est un stable si $\forall x, y \in X, xy \notin E(G)$, un stable X est maximal si $\forall y \in V \setminus X, X \cup \{y\}$ n'est pas un stable, un stable X est maximum si $\forall Y \subset V(G)$ stable, $|X| \geq |Y|$, le nombre de stabilité $\alpha(G)$ du graphe G est la taille d'un stable maximum.

Un graphe simple $G = (V, E)$ est complet si $E = [V]^2$, si $|V| = n$ et G est complet, on note alors $G \equiv K_n$, dans le cas d'un graphe complet K_n , le nombre d'arêtes est $\frac{n(n-1)}{2}$.

1.2.2 Chaînes, cycles, connexité, trous et antitrous

Soit G un graphe, on appelle chaîne de G toute suite $C = (v_1, v_2, \dots, v_k)$ ($k \geq 2$) de sommets de G vérifiant pour tout $1 \leq i \leq k-1 : v_i v_{i+1} \in E(G)$.

Les arêtes de G de la forme $v_i v_{i+1}$ avec $1 \leq i \leq k-1$, sont alors appelées les arêtes de la chaîne. Les arêtes de G de la forme $v_i v_j$ avec $|i-j| > 1$ sont appelées les cordes de la chaîne. Les sommets v_1 et v_k sont appelés les extrémités de la chaîne. La longueur de la chaîne est égale à son nombre d'arêtes.

Un cycle est une chaîne dont les extrémités coïncident.

Un graphe est dit connexe lorsqu'il existe une chaîne entre deux sommets quelconques de ce graphe. Un sous-graphe connexe maximal d'un graphe quelconque est une composante connexe de ce graphe.

Un graphe est k -connexe s'il reste connexe après retrait de $k-1$ sommets quelconques.

Si $G = (V, E)$ est un graphe connexe, alors : Un sommet $v \in V$ est un point d'articulation

si $G \setminus \{v\}$ n'est pas connexe.

Une arête $e \in E$ est un isthme si $G \setminus \{e\}$ ($G \setminus \{v\}$ désigne le graphe G privé de e et des arêtes incidentes à e) n'est pas connexe. Un bloc de G est soit un isthme, soit une composante 2-connexe (c'est à dire un sous graphe 2-connexe maximal).

Un trou est un cycle sans cordes. Un antitrou est le complémentaire d'un trou.

1.2.3 Quelques graphes particuliers

Nous présentons ici rapidement quelques classes de graphes particulières qui occupent une place majeure en théorie des graphes et que nous rencontrerons souvent dans la suite de ce travail, avec leurs propriétés et les liens entre elles.

Les graphes bipartis

Définition 1 *un graphe est dit biparti s'il existe une partition de son ensemble de sommets en deux sous-ensembles V_1 et V_2 telle que chaque arête ait une extrémité dans V_1 et l'autre dans V_2 .*

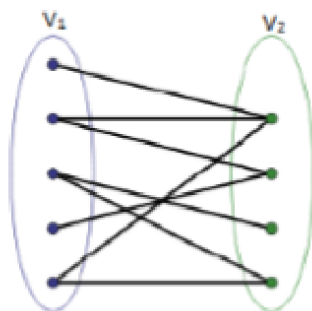


FIGURE 1.1 – Exemple d'un graphe biparti

Les graphes triangulés

Définition 2 *Un graphe simple $G = (V, E)$ est triangulé si tout cycle de longueur au moins 4 possède une corde, c'est-à-dire, une arête entre deux sommets non-consécutifs du cycle.*

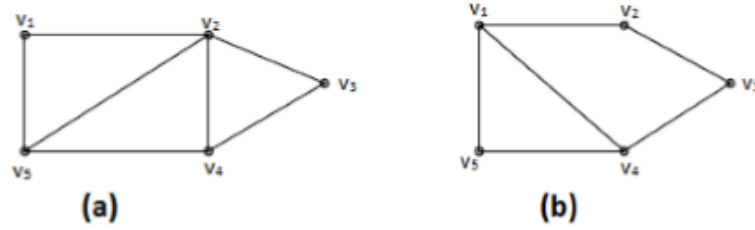
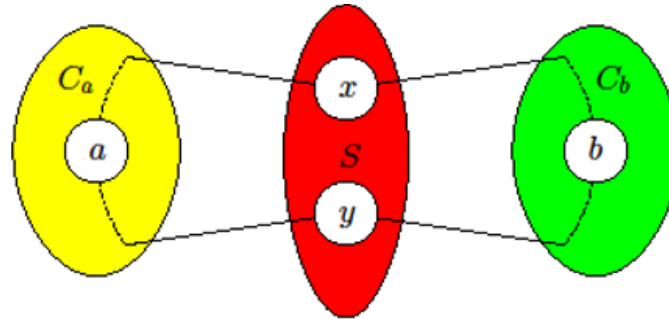


FIGURE 1.2 – (a-) un graphe triangulé (b-) un graphe qui n'est pas triangulé car il contient un cycle de longueur 4 qui n'a pas de corde.

Définition 3 Pour un graphe $G = (V, E)$, un séparateur est un sous-ensemble S de V tel que le sous-graphe induit par $V \setminus S$ n'est pas connexe, i.e. tel qu'il existe deux sommets $u, v \in V \setminus S$ qui ne sont pas reliés par une chaîne dans $G[V \setminus S]$, S est dit aussi un u, v -séparateur.

On dit que S est un séparateur minimal de G si S est minimal pour l'inclusion parmi tous les séparateurs de G .

Lemme 1 Soit G un graphe triangulé. Alors tout séparateur minimal de G est une clique.



Démonstration :

On suppose l'existence de $S \subseteq V$ séparateur minimal, et on considère deux sommets a, b non connectés dans $G[V \setminus S]$. Notons C_a (respectivement C_b) la composante connexe de $G[V \setminus S]$ contenant a (respectivement b).

Soient $x, y \in S$. Supposons par l'absurde x, y non-adjacents.

Alors x est adjacent à un sommet de C_a . En effet sinon x et a non connectés dans $G[V \setminus S']$, où $S' = S \setminus \{x\}$, et donc contradiction avec S minimal. De même x est adjacent à un sommet de C_b , et y est adjacent à un sommet de C_a et à un sommet de C_b . On considère alors un cycle de G de longueur minimale parmi ceux dont les sommets sont

dans $C_a \cup C_b \cup \{x, y\}$ et qui passent par x , par y , par un sommet de C_a et par un sommet de C_b (existe d'après ce qui précède). Un tel cycle est alors de longueur supérieure à 4. Or un tel cycle n'a pas de corde ; en effet, il est composé d'une chaîne de x à y qui est un chemin sans corde (par minimalité de la taille du cycle), et d'une chaîne de y à x qui est de même une chaîne sans corde. De plus il n'y a pas de corde entre un sommet du cycle appartenant à C_a et un autre appartenant à C_b (composantes connexes disjointes), et il n'y a pas de corde entre x et y par hypothèse. Finalement on a exhibé un trou de longueur supérieure à 4 : contradiction avec G triangulé.

Ainsi x et y sont adjacents et S est une clique.

Définition 4 *Un sommet u d'un graphe $G = (V, E)$ est dit simplicial si son voisinage $N_G(u)$ est une clique.*

Définition 5 *Un ordre d'élimination simplicial de G est une énumération v_1, v_2, \dots, v_n sur les sommets de G telle que, pour tout $1 \leq i \leq n$, v_i est un sommet simplicial dans le graphe $G_i = G[\mathcal{L}_i]$ où $\mathcal{L}_i = \{v_1, v_2, \dots, v_i\}$.*

Proposition 1 *La propriété d'être un graphe triangulé est une propriété héréditaire.*

Démonstration

Soit $G = (V, E)$ un graphe triangulé et soit $a \in V$. Notons $H = G[V \setminus \{a\}]$.

On suppose par l'absurde que H n'est pas triangulé : il contient un trou $v_0 \dots v_k$ de longueur $k \geq 4$. Il est alors clair que $v_0 \dots v_k$ est aussi un trou pour G : contradiction avec G triangulé.

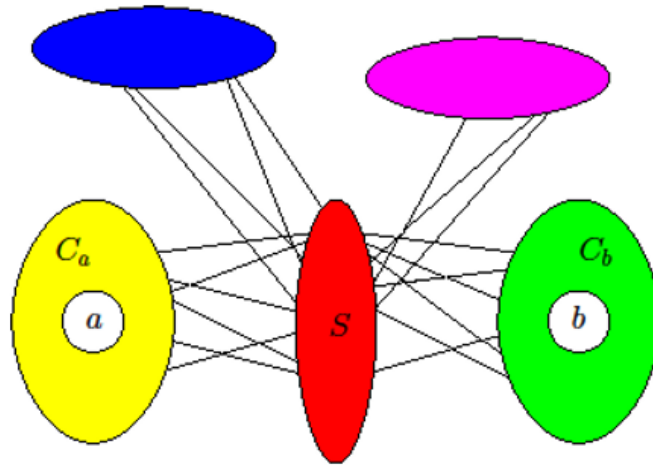
On voit ainsi que tout sous-graphe induit d'un graphe triangulé est aussi triangulé.

Lemme 2 *Soit G un graphe triangulé. Alors G admet au moins un sommet simplicial.*

Démonstration

On démontre par récurrence sur l'ordre de G la propriété suivante : pour G graphe triangulé, soit G est complet (et donc tous ses sommets sont simpliciaux), soit il admet au moins deux sommets simpliciaux non adjacents.

- Pour $n = 1$ c'est évident.
 - Soit $G = (V, E)$ un graphe triangulé d'ordre $n > 1$. Supposons la propriété établie pour tout graphe triangulé d'ordre strictement inférieur à n . On suppose alors G non complet. Alors il existe deux sommets de G non adjacents, et G possède des ensembles séparateurs.
-



On considère alors un séparateur minimal S . Il vient que $G[V \setminus S]$ possède deux sommets a et b appartenant à des composantes connexes respectives C_a et C_b distinctes. On distingue alors plusieurs cas :

- $C_a \cup S$ est une clique. Alors, comme $N(a) \subseteq (C_a \cup S)$ (aucun sommet d'une autre composante connexe de $G[V \setminus S]$ n'est adjacent à a), il vient que $N(a)$ est une clique et que a est simplicial (dans G).

- Comme $|C_a \cup S| < n$ ($b \notin C_a \cup S$), on applique l'hypothèque de récurrence à $G[C_a \cup S]$, qui n'est pas complet : il existe deux sommets non adjacents $x, y \in C_a \cup S$ qui vérifient que $N(x) \cap (C_a \cup S)$ est une clique (x simplicial dans $G[C_a \cup S]$), et de même pour y .

On distingue à nouveau plusieurs cas :

- ★ $x \in C_a$. Alors comme précédemment $N(x) \subseteq (C_a \cup S)$, et donc $N(x) = N(x) \cap (C_a \cup S)$ est une clique, et x simplicial (dans S).

- ★ $y \in C_a$. Alors de même y simplicial (dans G).

- ★ Sinon $x, y \in S$. On a alors deux sommets non adjacents dans un séparateur minimal, contradiction avec S est une clique (cf. Lemme 1).

Finalement on a exhibé un sommet simplicial appartenant à C_a . On procède de même pour C_b afin de montrer l'existence d'un deuxième sommet simplicial de G non adjacent au premier.

Proposition 2 *Un graphe est triangulé si et seulement si il admet un schéma d'élimination simplicial.*

Démonstration

\Rightarrow On applique récursivement le résultat du Lemme 2 précédent :

On considère un graphe triangulé $G = (V, E)$. Il admet un sommet simplicial a . Alors d'après la proposition 1, $G[V \setminus \{a\}]$ est aussi triangulé. On peut donc répéter l'opération jusqu'à arriver au graphe (\emptyset, \emptyset) .

\Leftarrow On raisonne par contra-posée. Si un graphe G n'est pas triangulé, alors il possède un trou de longueur supérieure ou égale à 4. Or aucun des sommets d'un tel cycle ne peut être "éliminé simplicialement" : le premier sommet à éliminer aura forcément dans son voisinage dans le sous-graphe induit par les sommets restant deux sommets non adjacents (ses voisins dans le cycle).

Les graphes planaires

Définition 6 *Un graphe est planaire s'il est possible de le représenter dans le plan de sorte que les arêtes ne se coupent pas.*



FIGURE 1.3 – Exemple d'un graphe planaire

Le graphe à gauche paraît ne pas être planaire mais sa représentation à droite nous démontre le contraire. Attention à ne pas se fier à notre vue.

Définition 7 *Un graphe planaire découpe le plan en plusieurs régions.*

- Une face F d'un graphe est par définition une région du plan limitée par des arêtes et qui ne contient ni sommet, ni arête dans son intérieur.
- Deux faces sont dites adjacentes si leur contour contient au moins une arête commune ; deux faces qui ne se touchent que par un sommet ne sont pas adjacentes.
- Le degré d'une face F , noté $\deg(F)$ est égal à la longueur du cycle ou de la chaîne fermée qui limite F .

Exemple

On considère le graphe suivant :

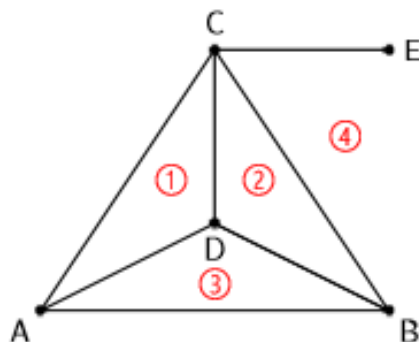


FIGURE 1.4 – Les faces d'un graphe planaire

Ce graphe comporte :

- la région (1) de degré 3 limitée par le cycle $A - C - D$;
- la région (2) de degré 3 limitée par le cycle $B - C - D$;
- la région (3) de degré 3 limitée par le cycle $A - B - D$;
- la région (4) de degré 5 autour de la chaîne fermée $E - C - B - A - C - E$.

Nous remarquons ainsi que toute arête limitant deux faces, ou intérieure à une face, est comptée deux fois dans la chaîne fermée.

1.3 Les algorithmes

La notion d'algorithme est malaisée à définir formellement, mais cela est tout à fait possible ([6]) : un algorithme est un objet mathématique tout aussi clair qu'un nombre entier ou un graphe. La notion de rapidité d'un algorithme est très importante si l'on envisage des applications pratiques. De plus, depuis des temps très reculés, on sait que la recherche d'algorithmes rapides enrichit globalement les mathématiques pures - nous voulons dire ici pures d'algorithmes. Citons comme exemple l'algorithme qui se trouve dans le livre 7 des Éléments d'Euclide [9] pour le calcul du PGCD de deux entiers. Toutefois, jusqu'au milieu du XX^e siècle, la notion de rapidité d'un algorithme était vague et ne faisait pas l'objet de recherches pour elle-même. Finalement le développement de la théorie de la complexité algorithmique depuis le milieu des années 1960 a placé la notion de rapidité d'un algorithme au cœur de questions très difficiles et profondes.

1.3.1 Problèmes, algorithmes, complexité

Un problème est défini par la donnée d'un objet mathématique représenté par un nombre fini de symboles (l'instance ou l'entrée) et d'une question dont la réponse dépend uniquement de l'instance. Un problème de décision est un problème dont la réponse est OUI ou NON.

Un algorithme est une description de longueur finie, dans un langage de programmation, d'une suite finie d'opérations élémentaires (le calcul) ne dépendant que de l'instance et fournissant une réponse à la question (la sortie). Tous nos algorithmes auront pour instance des graphes, ou des graphes avec quelques objets supplémentaires.

Pour chaque algorithme, et pour chaque entier n , nous chercherons à évaluer la complexité de l'algorithme, c'est-à-dire le nombre d'opérations élémentaires nécessaires à l'exécution de l'algorithme pour une instance de taille n (On appelle taille de l'instance le nombre de symboles nécessaires à sa représentation) dans le pire des cas. Nos estimations de complexité seront toujours données à une constante multiplicative près à l'aide de la notation de Landau : $O(\dots)$. Dire qu'un algorithme est de complexité $O(f(n))$ où f est une fonction mathématique, s'il existe une constante c telle que le nombre d'opérations nécessaires pour exécuter l'algorithme sur une instance de taille n est inférieur à $c \times f(n)$. Un algorithme est dit polynomial si la fonction f est un polynôme fonction de n , c'est-à-dire s'il existe un entier k tel que l'algorithme est de complexité $O(n^k)$. Dans ce cas, l'algorithme est considéré comme efficace.

La complexité d'un algorithme polynomial dont l'entrée est un graphe s'exprimera sous la forme $O((P(n, m)))$, où P est un polynôme fonction du nombre n de sommets et du nombre m d'arêtes. Le nombre maximum d'arêtes d'un graphe sur n sommets est $\frac{n(n-1)}{2}$. Il est donc possible d'exprimer une complexité fonction de n et m uniquement en fonction de n : un algorithme de complexité $O((P(n, m)))$ est aussi de complexité $O((P(n, n^2)))$.

1.3.2 Les problèmes P , NP , NP -complets et NP -difficiles

- **La classe P** : c'est la classe des problèmes pouvant être résolus en un temps polynomial (c'est-à-dire qu'il existe un algorithme en temps polynomial pour les résoudre). C'est la classe des problèmes dits faciles.

- **La classe NP** : Son nom est trompeur : il ne signifie pas "non polynomial", mais "polynomial non-déterministe". On dit qu'un problème est dans NP s'il peut être résolu par une machine non-déterministe en un temps polynomial. De façon équivalente, ce sont des problèmes dont on sait vérifier une solution avec une machine déterministe en un temps polynomial (on peut donc résoudre un tel problème en énumérant toutes les solutions, puis en les testant les unes après les autres ...). La classe NP est donc une extension de la classe P .
- Un problème Γ est dit NP -Difficile si pour tout problème Π dans NP , il existe une machine déterministe qui permette de ramener la résolution de Π à la résolution de Γ en un temps polynomial (on imagine bien dans ce cas que la résolution de Γ est donc "au moins aussi difficile" que la résolution de Π , sinon Π ne serait pas un problème de la classe NP).
- Un problème est dit NP -Complet s'il est NP -Difficile et appartient à la classe NP .

1.4 Conclusion

Dans ce chapitre, nous avons rappelé les notions de base de la théorie de la complexité et de théorie des graphes.

COLORATION DES GRAPHS

2.1 Introduction

On va traiter dans ce chapitre le problème de la coloration. Dans la première partie de ce chapitre, on va s'intéresser à la coloration des sommets, on va commencer par donner quelques définitions et propriétés, ensuite, présenter quelques résultats concernant la coloration de quelques graphes particulières. La deuxième partie est consacrée à la coloration des arêtes et ses propriétés. Mais avant ces deux parties, on va parler un peu de l'histoire et les domaines d'applications de la coloration des graphes.

2.2 Un peu d'histoire

Les premiers résultats de coloration de graphes concernent presque exclusivement la coloration des cartes. En cherchant à mettre en couleurs une carte des comtés d'Angleterre, Francis Guthrie (mathématicien et botaniste sud-africain 1831 – 1899) postule en 1852 la conjecture des quatre couleurs : il remarqua en effet qu'il n'y avait besoin que de quatre couleurs pour que deux comtés ayant une frontière commune soient de couleurs différentes. En 1879, Alfred Kempe (mathématicien anglais 1849 – 1922) publia ce qu'il prétendit en être une démonstration et pendant une décennie, on crut que le problème des quatre couleurs était résolu.

En 1890, Percy John Heawood (mathématicien anglais 1861 – 1955) fit remarquer que la démonstration de Kempe était fausse. Il montra quant à lui le théorème des cinq couleurs en reprenant des idées de Kempe.

De nombreux travaux ont été publiés lors du siècle suivant pour réduire le nombre de

couleurs à quatre, jusqu'à la démonstration finale de Kenneth Appel (mathématicien américain né en 1932) et Wolfgang Haken (mathématicien allemand né en 1928). Il s'agit aussi de la première preuve majeure utilisant massivement l'ordinateur.

2.3 Domaines d'applications

Nous avons vu que la coloration d'un graphe est utile dans la coloration des cartes géographiques mais ce n'est pas là son unique domaine d'application. En effet, la coloration de graphe a de nombreuses utilités :

- les problèmes d'incompatibilité : le stockage de produits chimiques qui peuvent exploser s'ils entrent en contact, désignation d'un endroit pour des personnes ou des animaux en tenant compte des relations,
- l'allocation de fréquences, par exemple dans un réseau de téléphone mobile GSM,
- la confection d'horaires,
- la résolution du Sudoku : ce jeu est ni-plus-ni-moins qu'un problème de coloration avec $k = 9$ couleurs,
- ...

2.4 Coloration des sommets

2.4.1 Définitions et propriétés

Définition 8 *Colorer un graphe, c'est associer une couleur à chaque sommet de façon que deux sommets adjacents soient colorés avec des couleurs différentes. Plus formellement, Une coloration d'un graphe $G = (V, E)$ est une application $c : V \rightarrow S$, où les éléments de S sont appelés les couleurs. Une coloration c est propre si pour toute arête $uv \in E(G)$, $c(u) \neq c(v)$. Comme la taille de S est le paramètre important, on considère souvent que $S = \{1, 2, \dots, k\}$. Une k -coloration d'un graphe G est une coloration propre de G à valeurs dans $S = \{1, 2, \dots, k\}$. Un graphe est k -colorable s'il admet une k -coloration. Le plus petit entier k tel que G soit k -colorable est le nombre chromatique de G , noté $\chi(G)$.*

On peut définir aussi une k -coloration comme une partition de l'ensemble des sommets en k parties stables.

Remarques

- La coloration d'un graphe n'est bien évidemment pas unique : d'une part les couleurs peuvent être changées (remplacer du noir par du rouge par exemple), d'autre part la répartition des couleurs peut varier, ainsi que le nombre de couleurs utilisées.
- Un graphe d'ordre n pourra toujours être coloré de n couleurs différentes.

Proposition 3 Soit $G = (V, E)$ un graphe simple d'ordre n : On a l'encadrement suivant :

$$\left\lceil \frac{n}{\alpha(G)} \right\rceil \leq \chi(G) \leq \Delta(G) + 1$$

où $\Delta(G)$ est le degré maximal des sommets du graphe G et $\alpha(G)$ est le cardinal de la plus grande partie stable (le nombre de stabilité de G).

Démonstration

Notons $\{S_1, S_2, \dots, S_k\}$ une partition de V en k parties stables avec $k = \chi(G)$. Alors :

$$n = \sum_{i=1}^k |S_i| \leq \sum_{i=1}^k \alpha(G) = k \cdot \alpha(G) = \chi(G) \cdot \alpha(G)$$

n , $\alpha(G)$ et $\chi(G)$ étant des entiers naturels, l'inégalité de gauche est établie.

Construisons une partition de V en sous-ensembles stables de la manière suivante :

- on considère un sommet v_1 arbitraire de V ; et S_1 est une plus grande partie stable de V contenant v_1 .
- s'il existe un sommet v_2 de V qui n'est pas dans S_1 , on construit S_2 ; une plus grande partie stable de V contenant v_2 disjointe de S_1 .
- s'il existe un sommet v_3 de V qui n'est pas dans $S_1 \cup S_2$; on construit S_3 ; plus grande partie stable de V contenant v_3 telle que S_1 , S_2 et S_3 soient deux à deux disjointes.
- V étant un ensemble fini, ce procédé se terminera et nous obtenons une partition $\{S_1, S_2, \dots, S_k\}$ de V : En choisissant une couleur par élément de la partition, nous aurons nécessairement : $\chi(G) \leq k$.

Considérons à présent un sommet v de la partie S_k : Le caractère maximal des parties construites assure que ce sommet v est adjacent à au moins un sommet de chaque partie S_i , $i \in \{1, 2, \dots, k-1\}$: On en déduit alors que $d(v) \geq k-1$, d'où : $\Delta(G) \geq d(v) \geq k-1 \geq \chi(G) - 1$

ce qui établit la deuxième inégalité.

Proposition 4 Soit $G = (V; E)$ un graphe simple d'ordre n ; alors :

$$\chi(G) + \alpha(G) \leq n + 1$$

Démonstration

Considérons S ; une partie stable de V de cardinal $\alpha(G)$: Une coloration possible des sommets consiste à colorier les sommets de S d'une même couleur et les $n - \alpha(G)$ autres sommets de couleurs toutes différentes. On en déduit que : $\chi(G) \leq 1 + (n - \alpha(G))$

Corollaire 1 Soit un graphe $G = (V, E)$.

$$\chi(G) = 1 \Leftrightarrow G \text{ est stable (n'a pas d'arêtes)}$$

$$\chi(G) = 2 \Leftrightarrow G \text{ est biparti}$$

◦ Quels sont les graphes avec un nombre chromatique exactement $\Delta + 1$?

Y'en a t-il beaucoup ?

Proposition 5 Soit G un graphe, si G est un cycle impair ou un graphe complet alors : $\chi(G) = \Delta(G) + 1$.

Démonstration

Si G est un cycle élémentaire de longueur impaire, alors on a $\chi(G) > 2$. Dans un cycle élémentaire, tous les sommets sont de degré 2, on sait que $\chi(G) \leq \Delta(G) + 1 = 3$. On en conclut que $\chi(G) = 3 = 2 + 1 = \Delta(G) + 1$.

Si G est un graphe complet alors tous ses sommets doivent être coloriés différemment et $\Delta(G) = |V| - 1$, donc $\chi(G) = |V| = \Delta(G) + 1$.

Théorème 2.4.1 (Brooks, 1941) Soit G un graphe connexe tel que $\Delta(G) \geq 3$ et G ne contient pas de clique à $\Delta(G) + 1$ sommets.

Alors $\chi(G) \leq \Delta(G)$.

Remarque

Si G est connexe, on peut numéroté les sommets de G de sorte que v_i à un voisin dans $\{v_1, \dots, v_{i-1}\}$. En effet, on choisit v_1 quelconque. Si on a déjà numéroté convenablement v_1, \dots, v_{i-1} , on choisit pour v_i l'extrémité d'une arête dans le cocycle $\omega(\{v_1, \dots, v_{i-1}\})$ (Soit A un ensemble de sommets d'un graphe, on appelle cocycle associé à A , qu'on notera

$\omega(A)$, l'ensemble des arcs incidents à A) et qui n'est pas dans cet ensemble $(\{v_1, \dots, v_{i-1}\})$. On a $\omega(\{v_1, \dots, v_{i-1}\}) \neq \emptyset$ sinon G ne serait pas connexe.

On va commencer par montrer le résultat dans un cas très particulier (le lemme), puis on généralisera.

Lemme 3 *S'il existe dans G deux sommets a et b tel que $d(a, b) = 2$ et si $G \setminus \{a, b\}$ est connexe, alors $\chi(G) \leq \Delta(G)$.*

Démonstration

Notons v_1 le voisin commun de a et b . D'après la remarque précédente, on peut numéroter les sommets de $V \setminus \{a, b\}$ sous la forme $\{v_1, \dots, v_{n-2}\}$ de sorte que v_i a un voisin dans $\{v_1, \dots, v_{i-1}\}$.

Montrons que l'on peut colorier le graphe avec $\Delta(G)$ couleurs : on colorie a et b avec la couleur 1. On colorie les sommets de v_{n-2} à v_1 en utilisant la plus petite couleur disponible. Pour tout sommet v_i (pour $2 \leq i \leq n-2$), v_i a un voisin dans v_1, \dots, v_{i-1} , la partie non encore coloriée. Comme $d(v_i) \leq \Delta(G)$, il reste une couleur disponible dans $\{1, \dots, \Delta(G)\}$. Lorsqu'on colorie v_1 , tous ses voisins sont déjà coloriés et a et b ont la même couleur : il reste une couleur disponible pour v_1 dans $\{1, \dots, \Delta(G)\}$.

Intéressons nous maintenant au cas général :

Démonstration

On raisonne sur la connexité de G .

Si G est 3-connexe : le graphe n'étant pas une clique (par hypothèse), il existe deux sommets a et b à distance 2. Comme G est 3-connexe, $G \setminus \{a, b\}$ est connexe et on peut appliquer le lemme.

Si G est 2-connexe : G ne peut pas avoir plus de 2 sommets universels (de degré $n-1$) puisque tant que l'on ne supprime pas tous les sommets universels, on ne déconnecte pas le graphe.

* Si G possède 1 ou 2 sommets universels : appelons u l'un d'entre eux. G n'étant pas complet (par hypothèse), il existe deux sommets a et b (différents de u , à distance 2 l'un de l'autre. Un chemin de longueur de 2 entre a et b passe par u car u est universel. $G \setminus \{a, b\}$ est connexe puisque l'on conserve le sommet universel u . On peut donc encore appliquer

le lemme.

* Si G ne possède pas de sommet universel : choisissons un sommet $u \in V$ tel que $3 \leq d(u) < n - 1$ (possible car $\Delta(G) \geq 3$). Le graphe $G \setminus \{u\}$ est soit 2-connexe, soit 1-connexe.

◦ Si $G \setminus \{u\}$ est 2-connexe : $\{u\} \cup \Gamma(u) \neq V$ car $d(x) < n - 1$. G étant connexe, $w(\{u\} \cup \Gamma(u)) \neq \emptyset$. Soit v l'extrémité d'une arête de ce cocycle, à l'extérieur de $\{u\} \cup \Gamma(u)$. u et v ne sont pas reliés donc v est relié à un voisin de u et $d(u, v) = 2$. De plus, $G \setminus \{u, v\}$ est connexe car $G \setminus \{u\}$ est 2-connexe. On peut donc appliquer le cas vu plus haut.

◦ Si $G \setminus \{u\}$ est 1-connexe : $G \setminus \{u\}$ admet au moins un point d'articulation. On considère $(G \setminus \{u\})_B$ l'arbre des blocs, il a au moins deux blocs pendants B_1 et B_2 (correspondant aux feuilles) puisque c'est un arbre, et il existe un point d'articulation. Ces deux blocs sont reliés au reste de $G \setminus \{u\}$ par des points d'articulation b_1 et b_2 . En fait, u a un voisin $a \neq b_1$ dans B_1 car sinon, en enlevant b_1 à G , on déconnecterait le graphe initial (absurde car G est 2-connexe). De même u a un voisin b dans B_2 autre que b_2 .

a et b ne sont pas reliés car sinon, on aurait deux chaînes disjointes les reliant (l'arête ab et la chaîne passant par b_1 puis b_2), et on contredirait le fait que a et b sont dans 2 blocs différents. De plus ils ont u comme voisin commun donc sont à distance 2 l'un de l'autre. Le graphe $G \setminus \{a, b, u\}$ est connexe car a et b ne sont pas les points d'articulation de B_1 et B_2 .

$G \setminus \{a, b\}$ est connexe car $d(u) \geq 3$ donc u est connecté à $G \setminus \{a, b\}$. Donc on peut appliquer à $G \setminus \{a, b\}$ le résultat vu plus haut.

Si G est 1-connexe : on considère B un bloc pendant de G . Soit b le point d'articulation de B .

* Si B est une clique à $\Delta(B) + 1$ sommets, $\Delta(B) + 1 \leq \Delta(G)$ puisque G ne contient pas de clique de taille $\Delta(G) + 1$. On peut donc colorier B à l'aide d'au plus $\Delta(G)$ couleurs.

* Si B n'est pas une clique à $\Delta(B) + 1$ sommets, on peut toujours colorier B en $\Delta(B) \leq \Delta(G)$ couleurs.

On a montré que l'on peut colorier B avec au plus $\Delta(G)$ couleurs. Reste à montrer le résultat pour tout G :

Par récurrence, on peut colorier le graphe $(G \setminus B) \cup \{b\}$ avec $\Delta(G)$ couleurs. En faisant coïncider les deux colorations sur b , on obtient une coloration de G en $\Delta(G)$ couleurs.

Si G est non-connexe : il suffit de travailler sur les composantes connexes.

Théorème 2.4.2 (Tutte, 1954) *Pour tout entier $k \geq 2$, il existe un graphe k -chromatique sans triangles (sans cliques à 3 sommets).*

Démonstration

On distingue trois cas.

- $k = 2$ C'est trivial car un graphe 2-chromatique est biparti et ne peut donc pas contenir de triangle.
- $k = 3$ Un cycle de longueur impaire convient (sans triangle).
- $k \geq 4$ On construit une famille de graphes $G_i = (V_i, E_i)$ telle que G_i est i -chromatique et sans triangle. Pour G_3 , on prend C_5 le cycle à 5 sommets. On note $n_i = |V_i|$.

Supposons G_i construit et construisons G_{i+1} : on prend $p = C_{in_i-i+1}^{m_i}$ copies de G_i notées G_{i_1}, \dots, G_{i_p} . On ajoute $in_i - i + 1$ sommets appelés centraux (ils forment un ensemble noté A). On note A_1, \dots, A_p les sous ensembles de A à n_i sommets.

Pour tout j tel que $1 \leq j \leq p$, on ajoute un couplage parfait entre G_{i_j} et A_j . Cela forme G_{i+1} . Montrons que cela convient :

G_i sans triangle $\Rightarrow G_{i+1}$ sans triangle : car s'il y a un triangle dans G_{i+1} , il contient un arête ab avec $a \in A$ et $b \in B$. Mais un sommet de A ne peut pas avoir 2 voisins dans une même copie de G_i (à cause du couplage parfait). Contradiction.

G_i i -coloriable $\Rightarrow G_{i+1}$ $(i+1)$ -coloriable : il suffit de reporter la i -coloration de G sur chaque copie, et d'ajouter une couleur pour a .

G_{i+1} i -coloriable $\Rightarrow G_i$ $(i-1)$ -coloriable : considérons une i -coloration de G_{i+1} . C'est en particulier une i -coloration de A . Notons $\{s_1, \dots, s_i\}$ la partition de A en classes de couleurs. L'un des s_j a au moins n_i sommets. En effet, supposons que $|s_j| < n_i$ (où $1 \leq j \leq i$), on aurait $|A| = \sum_1^i |S_j| \leq \sum_1^i (n_i - 1) = i(n_i - 1) < in_i - i + 1 = |A|$. Contradiction.

Il existe donc un sous-ensemble A_k de A de cardinal n_i monochromatique. Le graphe G_{i_k} couplé avec A_k ne peut pas utiliser cette couleur. On en déduit que G_{i_k} est $(i-1)$ -colorié donc G_i est $(i-1)$ -coloriable. Par récurrence, on en déduit que G_{i+1} est $(i+1)$ -chromatique.

2.4.2 Complexité

Décider si un graphe possède une coloration en k couleurs est un problème NP -complet [4], et trouver le nombre chromatique $\chi(G)$ d'un graphe est un problème NP -difficile. Le problème du nombre chromatique devient polynomial si les sommets du graphe ont un degré d'au plus 2 ou si le graphe est biparti.

2.4.3 Coloration des graphes planaires

Théorème des 4 couleurs

Théorème 2.4.3 (*Appel, Haken 1976*) *Tout graphe planaire $G = (V, E)$ est 4-coloriable.*

Historique :

- 1852 : conjecturé par Guthrie.
- 1879 : démonstration de Kempe, mais erronée (Heawood 1890).
- 1880 : démonstration de Tait, mais erronée (Petersen 1891).
- 1976 : démonstration de Appel et Haken, assistée par ordinateur (1476 cas critiques, 1200 heures de calcul).
- 1997 : démonstration de Robertson, Sanders, Seymour, Thomas, toujours assistée par ordinateur (633 cas).
- 2005 : démonstration formalisée en Coq par Gonthier et Werner.

2.4.4 Coloration des graphes parfaits

Soit G un graphe, la taille d'une clique de taille maximum de G est notée $\omega(G)$. Il est clair que si un graphe G contient une clique de taille k , il faut au moins k couleurs pour le colorier. Donc, tout graphe G vérifie : $\omega(G) \leq \chi(G)$. Il est naturel de se demander s'il y a égalité entre le nombre chromatique et la taille d'une plus grande clique. Ce n'est pas le cas et le plus petit contre-exemple est C_5 (le graphe C_k désigne un trou sur k sommets) : $\chi(C_5) = 3$ et $\omega(C_5) = 2$. Plus généralement, n'importe quel trou impair vérifie $\chi(C_{2k+1}) = 3$ et $\omega(C_{2k+1}) = 2$. Ghouila-Houri [2] a également remarqué que n'importe quel antitrou impair vérifie $\chi(\overline{C_{2k+1}}) = k+1$ et $\omega(\overline{C_{2k+1}}) = k$. Donc, l'égalité $\chi(G) = \omega(G)$ n'est pas vérifiée en général, mais il est intéressant de se demander pour quels graphes on

a égalité. Dans un graphe G , une coloration de G et une clique de G sont dites de même taille si la taille de la clique est exactement égale au nombre de couleur de la coloration, dans ce cas, la clique est de taille $\omega(G)$ et la coloration utilise $\chi(G)$ couleurs.

En 1960, Berge [4] a défini la classes des graphe parfaits comme étant la classe des graphes G tels que tout sous-graphe induit H de G vérifie $\chi(H) = \omega(H)$. Cette classe de graphes est apparue comme étant une classe de graphes assez générale sur laquelle le problème de la coloration pouvait être résolu efficacement. En 1984, Grötschel, Lovász et Schrijver, [17] ont donné un algorithme polynomial qui trouve, pour tout graphe parfait, une coloration et une clique de même taille. Déterminer le nombre chromatique d'un graphe est NP -complet en général, mais devient donc polynomial sur la classe des graphes parfaits.

Les trous impairs et les antitrous impairs ne sont pas des graphes parfaits. Ainsi, dès qu'un graphe contient l'un d'eux, il n'est pas non plus parfait. Les graphes sans trou impair et sans antitrou impair sont donc la plus grande classe de graphes dont il est possible d'espérer la perfection. Un graphe est dit de Berge si et seulement s'il ne contient ni trou impair, ni antitrou impair.

Au début des années 1960, Berge a énoncé deux conjectures célèbres [5]. La première, dite conjecture faible des graphes parfaits, a été étudiée par par Fulkerson [8], puis finalement démontrée par Lovász en 1971 [13, 14]. C'est ce que nous appelons maintenant le théorème des graphes parfaits :

Théorème 2.4.4 ([13, 14]) *Un graphe est parfait si et seulement si son complémentaire est parfait.*

La deuxième conjecture de Berge, dite conjecture forte des graphes parfaits, a fait l'objet de nombreuses recherches avant d'être finalement démontrée par Chudnovsky, Robertson, Seymour et Thomas en 2002 [16]. C'est ce que nous appelons maintenant le théorème fort des graphes parfaits :

Théorème 2.4.5 ([16]) *Un graphe est parfait si et seulement s'il est de Berge.*

Le problème de la reconnaissance des graphes parfaits a lui aussi été résolu. En 2002, Chudnovsky, Cornuéols, Liu, Seymour et Vušković [15] ont donné un algorithme de complexité $O(n^9)$ permettant de reconnaître les graphes parfaits.

2.5 Coloration des arêtes

2.5.1 Définitions et propriétés

On peut aussi colorier les arêtes d'un graphe pour que deux arêtes incidentes à un même sommet n'aient pas la même couleur.

Définition 9 *Soit G un graphe. Une coloration des arêtes de G est une application $\gamma : E \rightarrow C$ de l'ensemble des arêtes de G dans un ensemble fini C de couleurs, telle que si un sommet x est une extrémité des arêtes e et e' alors $\gamma(e) \neq \gamma(e')$.*

On cherche généralement à déterminer une coloration utilisant aussi peu de couleurs que possible. Le plus petit nombre de couleurs nécessaires pour colorer les arêtes d'un graphe G s'appelle : l'indice chromatique de G et est noté $\chi'(G)$. On peut définir aussi l'indice chromatique comme le cardinal minimal d'une partition en couplages (Un couplage dans un graphe est un ensemble d'arêtes n'ayant aucune extrémité en commun) de l'ensemble des arêtes.

Exemple

Le graphe suivant a un indice chromatique de 4.

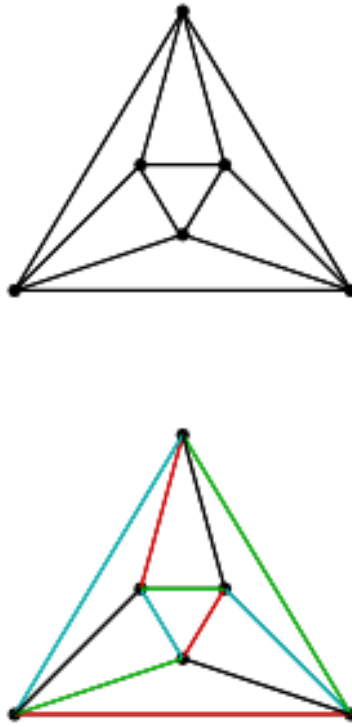


FIGURE 2.1 – Exemple de coloration d'arêtes d'un graphe

Remarque

Le problème de coloration des arêtes peut se ramener à celui des sommets pour un graphe en définissant le graphe adjoint G^a ainsi : les sommets de G^a sont les arêtes de G et deux sommets de G^a sont adjacents si, dans G ces deux arêtes ont une extrémité communes.

Proposition 6 Soit $G = (X, E)$ un graphe. On note $\nu(G)$ la taille maximale d'un couplage (la taille d'un couplage de taille maximum).

$$\Delta(G) \leq \chi'(G) \quad \left\lceil \frac{|E|}{\nu(G)} \right\rceil \leq \chi'(G)$$

Démonstration

Tout d'abord, il existe un sommet v tel que $d(v) = \Delta(G)$. En v , on a besoin de $\Delta(G)$ couleurs différentes donc $\Delta(G) \leq \chi'(G)$.

Ensuite, soit $\{C_1, \dots, C_k\}$ une partition de E minimale, en couplage. On a $k = \chi'(G)$.

$$|E| = \sum_{i=1}^k |C_i| \leq \sum_{i=1}^k \nu(G) = \chi'(G) \cdot \nu(G)$$

Donc $\chi'(G) \geq \frac{|E|}{\nu(G)}$. Le fait que $\chi'(G)$ soit un entier permet d'ajouter la partie entière

supérieure.

Proposition 7

$$\chi'(K_{2n}) = 2n - 1 \qquad \chi'(K_{2n+1}) = 2n + 1$$

Démonstration

Le cas de K_{2n} $m(K_{2n}) = \frac{2n(2n-1)}{2} = n(2n-1)$ et on voit facilement que $\nu(K_{2n}) = n$.

La proposition précédente donne alors $\chi'(K_{2n}) \geq 2n - 1$.

Pour montrer l'inégalité contraire, on colorie K_{2n} avec $2n - 1$ couleurs. On numérote tout d'abord les sommets de 0 à $2n - 1$. On place $2n - 1$ au centre tandis que les $2n - 1$ autres forment un polygone régulier tout autour.

On relie alors $2n - 1$ à 0, puis 1 à $2n - 2$, 2 à $2n - 3$, ..., et $n - 1$ à n . On forme ainsi un couplage parfait C_0 . On décale les sommets du polygone pour former le couplage parfait C_1 . $2n - 1$ est relié à 1, puis 2 à 0, 3 à $2n - 2$, 4 à $2n - 3$, ..., n à $n + 1$.

De la façon, on forme successivement les couplages parfaits C_0 à C_{2n-2} . Le couplage C_k est en fait $\{(k, 2n - 1)\} \cup \{(i + k, 2n - 1 - i + k) \mid 1 \leq i \leq n - 1\}$ (on compte les sommets modulo $2n - 1$).

Il est clair que ces $2n - 1$ couplages forment une partition de l'ensemble des arêtes. En associant une couleur à chaque couplage, on obtient le coloriage recherché.

Le cas de K_{2n+1} $m(K_{2n+1}) = \frac{2n(2n+1)}{2} = n(2n+1)$ et on voit que $\nu(K_{2n+1}) = n$. La proposition précédente donne alors $\chi'(K_{2n+1}) \geq 2n + 1$.

K_{2n+1} est inclus dans K_{2n+2} donc $\chi'(K_{2n+1}) \leq \chi'(K_{2n+2}) = 2n + 1$.

Théorème 2.5.1 (Vizing, 1964) Pour tout graphe G , $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$.

Démonstration

Il nous reste à prouver la seconde inégalité. On va raisonner par récurrence sur $|E|$. Si $|E| \leq \Delta + 1$, il est clair qu'on peut colorier le graphe avec $\Delta + 1$ couleurs.

Si $|E| > \Delta + 1$, on peut supposer que tout le graphe sauf une arête se colorie avec $\Delta + 1$ couleurs. Notons xy_1 cette arête non-coloriée.

La définition de $\Delta(G)$ impose l'absence d'au moins une couleur en chaque sommet. En effet, les degrés des sommets sont au maximum Δ et on utilise ici $\Delta + 1$ couleurs.

S'il manque une même couleur en x et y_1 , on colorie l'arête avec cette couleur et le problème est résolu. Tout le problème réside donc dans le cas où les couleurs manquantes en

x et en y_1 sont différentes.

On construit la suite $(xy_1, C_1), (xy_2, C_2), \dots$ et (xy_h, C_h) vérifiant C_i est la couleur manquante en y_i et l'arête y_{i+1} est coloriée en C_i . On prend cette suite maximale. Cela impose soit l'absence d'une arête incidente à x de couleur C_h , soit la présence de C_h parmi les éléments précédents de la suite.

On note C la couleur absente en x . Par construction, $C \neq C_i$ pour tout $1 \leq i \leq k-1$.

Deux cas se présentent :

Si C_h est absente en x , on décale toutes les couleurs : xy_1 est colorié en C_1, \dots, xy_h en C_h .

Le problème est alors résolu.

Sinon, la couleur C_h est présente en x . Par maximalité de la suite, il existe j tel que xy_j est colorié en C_h ($C_{j-1} = C_h$).

On décale également les couleurs : xy_i est colorié en C_i pour $i < j$, xy_i rest en C_{i-1} pour $i > j$. xy_j n'a plus de couleur.

On considère le graphe partiel H de G réduit aux couleurs C et C_h . On a $d_H(x) = 1$ car C est absente mais C_h est présente. $d_H(y_h) \leq 1$ car C_h est absente. $d_H(y_j) \leq 1$ car on a supprimé la couleur C_h .

Si x et y_j sont dans des composantes connexes de H différentes. Si $d_H(y_j) = 0$, c'est-à-dire C absente en y_j , il suffit de colorier xy_j en C pour résoudre le problème.

Sinon, $d_H(y_j) = 1$ et C est présente en y_j . La composante connexe de x est une chaîne alternée de C et C_h finissant en x (grâce au degré) par une arête C_h . La composante de y_j finit en y_j par une arête C . On échange donc C et C_h dans cette dernière et on colorie xy_j en C pour résoudre le problème.

Si x et y_j sont dans la même composante connexe de H . On a fini de décaler les couleurs dans le graphe initial. xy_i est colorié en C_i pour tout $1 \leq i \leq h-1$ et xy_h n'est pas coloriée. On considère de nouveau le graphe partiel H . Il n'a pas changé car on a décalé des couleurs différentes de C et C_h . On a toujours une chaîne entre y_j et x , alternée en C et C_h , finissant par une arête C_h en x . La composante connexe de y finit en y par une arête coloriée en C . On échange C et C_h dans cette composante pour résoudre le problème.

Remarque

Déterminer si $\chi'(G)$ vaut $\Delta(G)$ ou $\Delta(G) + 1$ est un problème *NP*-complet.

Théorème 2.5.2 (König 1916) Si G est un graphe biparti alors $\chi'(G) = \Delta(G)$.

Démonstration Par récurrence sur le nombre d'arêtes. Si $E(G) = 0$, le théorème est trivialement vrai. Supposons maintenant que $E(G) \geq 1$ et que le résultat est vrai pour les graphes ayant moins d'arêtes. Posons $\Delta(G) = \Delta$. Soit xy une arête de G . Par hypothèse de récurrence, $G - xy$ admet un Δ -coloration.

Dans $G - xy$, les sommets x et y sont chacun incident à au plus $\Delta - 1$ arêtes. Ainsi il existe $c_x, c_y \in \{1, 2, \dots, \Delta\}$ tels que x (resp. y) ne soit pas adjacent à une arête colorée c_x (resp. c_y). Si $c_x = c_y$, alors on peut colorer xy avec cette couleur et on obtient une Δ -coloration de G . Nous pouvons donc supposer que $c_x \neq c_y$ et que le sommet x est incident à une arête e colorée c_y .

Étendons cette arête en une marche maximale P dont les arêtes sont colorées c_y et c_x alternativement. Comme un sommet est adjacent à au plus une arête d'une couleur, P est un chemin. De plus, P ne contient pas y . Sinon P terminerait en y avec une arête colorée c_x et P serait un chemin pair. Mais alors $P + xy$ formerait un cycle impair, ce qui est impossible. On peut donc recolorer P en intervertissant les couleurs c_x et c_y . Par maximalité de P , deux arêtes adjacents sont toujours de couleurs différentes. En colorant xy par c_y , on obtient alors une Δ -coloration de G .

2.5.2 Complexité

Le calcul de l'indice chromatique d'un graphe est un problème *NP*-difficile. Soit Δ le degré maximal des sommets d'un graphe G . Il est clair que Δ est une borne inférieure simple de l'indice chromatique $\chi'(G)$. L'indice chromatique ne peut prendre que deux valeurs Δ ou $\Delta + 1$. On connaît des algorithmes polynomiaux pour trouver une coloration des arêtes en au plus $\Delta + 1$ couleurs, mais l'existence d'une Δ -coloration est un problème *NP*-complet.

2.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la coloration des sommets et des arêtes et leurs propriétés.

QUELQUES ALGORITHMES DE COLORATION

3.1 Introduction

La coloration des graphes quand l'ordre est petit est assez aisée, mais le problème se complique dès que le nombre de sommets augmente, donc pour colorier un graphe, des algorithmes ont été développés.

Les algorithmes que nous allons présenter (il en existe d'autres) permet d'obtenir une assez bonne coloration d'un graphe, c'est-à-dire une coloration n'utilisant pas un trop grand nombre de couleurs.

3.2 Graphes quelconques : des algorithmes d'approximation

3.2.1 Coloration séquentielle : l'algorithme glouton

Une première approche pour colorer le graphe est de prendre ses sommets les uns après les autres afin de leur affecter une couleur, tout en veillant à ce que deux sommets adjacents n'aient jamais la même couleur : c'est l'algorithme de coloration séquentielle. On parle d'algorithme glouton dans le sens où il consiste à faire le choix d'une couleur "optimum" localement, dans l'espoir d'obtenir une coloration finale "satisfaisante".

On peut par exemple décider de prendre pour chaque sommet la plus petite couleur non utilisée dans les voisins du sommet à colorier. L'algorithme fourni ainsi une borne supérieure pour $\chi(G)$, c'est-à-dire une coloration propre, mais pas forcément optimale. La

coloration obtenue dépendra fortement de l'ordre dans lequel les sommets seront colorés. Notons ainsi que pour un graphe donné il existe toujours un ordre des sommets pour lequel l'algorithme fournisse une coloration optimale : il suffit de prendre une coloration optimale, et de numéroter les sommets par ordre croissant de couleur. Malheureusement c'est un "cerce vicieux", et l'on ne peut pas connaître à l'avance une telle numérotation. Dans ce qui suit, on considère un graphe $G = (V, E)$ à colorer, et on notera n (resp. m) l'ordre (resp. la taille) de G . Les sommets du graphe seront alors des entiers numérotés de 0 à $n - 1$: $V = [0, n - 1]$. Les numérotations ou les ordres considérés sont alors des permutations de $[0, n - 1]$ dans lui-même. De plus, si on ne le précise pas, lorsqu'il sera question de la complexité d'un algorithme, il s'agira généralement d'une complexité temporelle.

Algorithme 1 Coloration séquentielle

Entrée - Un graphe quelconque $G = (V, E)$

-Une permutation des sommets $\sigma : [0, n - 1] \rightarrow V$

Sortie - Une coloration propre $c : V \rightarrow \mathbb{N}$

```

1 :    $n := |V|$ ;
2 :   Couleur := Tableau (Taille :  $n$ ) (Défaut :  $-1$ );
3 :   Pour  $i$  de 0 à  $n - 1$  faire
4 :        $x := \sigma(i)$ ;
5 :       # Recherche de la plus petite couleur non utilisée dans  $\mathcal{N}(x)$  #
6 :       Libre := Tableau (Taille : ???) (Défaut : vrai);
7 :       Pour  $y$  dans  $\mathcal{N}(x)$  faire
8 :           Si Couleur. $(y) \neq -1$  alors
9 :               Libre.(Couleur. $(y)) \leftarrow faux$ ;
10 :          Fin Si
11 :      Fin Pour;
12 :       $index := 0$ ;
13 :      Tant Que Libre. $(index) = faux$  faire
14 :           $index := index + 1$ 
15 :      Fin Tant Que;
16 :      # Affecter la couleur donnée par index à  $x$  #
```

```

17 :      Couleur.(x) ← index
18 :      Fin Pour ;
19 :      Renvoyer Couleur

```

Proposition 8 *La complexité temporelle d'un algorithme de coloration séquentielle est : $O(m + n)$.*

Démonstration

On effectue une première boucle de n itérations. Pour chaque itération, la recherche de `index` (lignes 6 à 15) a une complexité temporelle $O(d_G(x))$, où $d_G(x) = |\mathcal{N}_G(x)|$ (nombre d'itérations de la boucle lignes 7 à 11). La complexité totale de ces opérations de recherche est donc dominée par $\sum_{x \in V} d_G(x) = m$. Finalement, on a bien une complexité temporelle totale $O(m + n)$.

3.2.2 Une première heuristique, Welsh & Powell

Il existe entre autres une méthode heuristique : l'algorithme de Welsh & Powell, il s'agit d'utiliser l'algorithme de coloration séquentiel mais cette fois-ci avec un ordre judicieux, afin d'obtenir une coloration propre la plus "acceptable" possible. L'algorithme de Welsh & Powell consiste ainsi à colorer séquentiellement le graphe en visitant les sommets par ordre de degré décroissant. L'idée est que les sommets ayant beaucoup de voisins seront plus difficiles à colorer, et donc il faut les colorer en premier. Cet algorithme couramment utilisé permet d'obtenir une assez bonne coloration d'un graphe, c'est-à-dire une coloration n'utilisant pas un trop grand nombre de couleurs, mais il n'assure pas que le nombre de couleurs utilisé soit minimum (et donc égal au nombre chromatique). On parle d'heuristique dans le sens où l'algorithme fournit une réponse approchée au problème de coloration.

Nous allons décrire ci-après l'algorithme de coloration de Welsh & Powell :

1. On classe d'abord les sommets du graphe dans l'ordre décroissant de leur degré. On obtient ainsi une liste x_1, x_2, \dots, x_n de sommets telle que $\deg(x_1) > \deg(x_2) > \dots > \deg(x_n)$.
 2. On choisit une couleur c_1 pour le sommet x_1 , et :
 - en parcourant la liste dans l'ordre, on attribue la couleur c_1 au premier sommet non colorié et non adjacent à x_1 ;
-

- en continuant à parcourir la liste dans l'ordre, on attribue la couleur c_1 aux autres sommets non coloriés et non adjacents aux sommets déjà coloriés avec c_1 et, ce, jusqu'à la fin de la liste.
3. S'il reste des sommets non coloriés, on attribue une nouvelle couleur au premier sommet non colorié et on reprend la démarche.
4. On s'arrête dès que tous les sommets ont été coloriés.

Algorithme 2 Welsh & Powell

Entrée - Un graphe quelconque $G = (V, E)$

Sortie - Une coloration propre $c : V \rightarrow \mathbb{N}$

```

1 :    $n := |V|$  ;
2 :    $\text{Deg} := \text{Tableau (Taille : } n \text{) (Défaut : 0)}$  ;
3 :    $\text{Sommet} := \text{Tableau (Taille : } n \text{) (Défaut : 0)}$  ;
4 :   Pour  $i$  de 0 à  $n - 1$  faire
5 :        $\text{Deg.}(i) \leftarrow d(i)$  ;
6 :        $\text{Sommet.}(i) \leftarrow i$  ;
7 :   Fin Pour ;
8 :    $\text{Tri (Tableau : Sommet) (Relation d'ordre : } a \leq b \text{ ssi } \text{Deg.}(a) \geq \text{Deg.}(b))$  ;
9 :   Coloration séquentielle (Graphe :  $G$ ) (Numérotation : Sommet)
```

Si on utilise par exemple un tri par dénombrement (en $O(n)$, car les degrés de deux sommets ne peuvent différer de plus de n), on obtient un algorithme de complexité $O(m + n)$. Si on utilise plutôt un tri par comparaison, l'algorithme est alors de complexité $O(n \ln(n) + n)$.

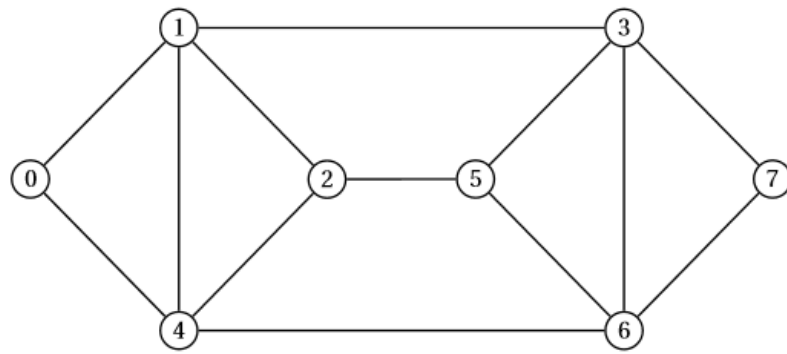
Exemple

Considérons le graphe G dessiné ci-dessous.

Appliquons l'algorithme décrit ci-dessus : 1, 3, 6, 4, 2, 5, 0, 7 est une liste des sommets classés dans l'ordre décroissant de leurs degrés.

D'après l'algorithme :

- à la première étape on attribue une couleur c_1 aux sommets 1 et 6 ;
- à la deuxième étape, on attribue une couleur c_2 aux sommets 3 et 4 ;
- à la troisième étape, on attribue une couleur c_3 aux sommets 2, 0 et 7 ;



- enfin à la dernière étape on attribue une couleur c_4 au sommet 5.

Un tel algorithme peut se présenter sous forme de tableau :

Sommets (degré)	1(4)	3(4)	6(4)	4(4)	2(3)	5(3)	0(2)	7(2)
étape 1	c_1		c_1					
étape 2		c_2		c_2				
étape 3					c_3		c_3	c_3
étape 4						c_4		

TABLE 3.1 – Tableau de coloration de graphe par welsh and powell

On obtient un coloriage de G avec 4 couleurs, mais en fait le nombre chromatique de G est 3.

on peut parfois aboutir aux pires colorations possibles ($\frac{n}{2}$ au lieu de 2). L'heuristique DSATUR propose une amélioration du principe de l'algorithme de Welsh & Powell afin d'éviter ce problème.

3.2.3 Deuxième heuristique, DSATUR

DSATUR est un algorithme de coloration de graphes créé par Daniel Brélaz en 1979 à l'EPFL (École polytechnique fédérale de Lausanne).

Cette fois, on considère toujours qu'un sommet est plus difficile à colorer s'il a beaucoup de voisins, mais on prend également en compte le fait qu'un sommet qui possède déjà beaucoup de voisins de couleurs différentes sera aussi problématique. On modifie alors l'algorithme de coloration séquentielle pour colorer en priorité de tels sommets. Pour chaque sommet v de V , on définit le degré de saturation $DSAT(v)$ calculé à un certain

moment de l'algorithme de la manière suivante :

Si aucun voisin de v n'est colorié alors

$$DSAT(v) = \text{degré}(v) .$$

Sinon

$$DSAT(a) = \text{nombre de couleurs différentes utilisées pour colorer les voisins de } a .$$

L'algorithme consiste alors à colorer séquentiellement le graphe en prenant à chaque étape un sommet non colorié de degré de saturation maximale, où on impose de prendre un sommet de degré maximal en cas de conflit. Il ne faudra donc pas oublier de mettre à jour à chaque étape le degré de saturation d'un sommet. On remarque également que lorsqu'on colore un sommet a , il n'est nécessaire de mettre à jour $DSAT(a)$ que pour les sommets voisins de a .

Algorithme 3 DSATUR

Entrée - Un graphe quelconque $G = (V, E)$

Sortie - Une coloration propre $c : V \rightarrow \mathbb{N}$

```

1.     $n := |V|$  ;
2.    Couleur := Tableau (Taille :  $n$ ) (Défaut :  $-1$ ) ;
3.    Deg := Tableau (Taille :  $n$ ) (Défaut :  $0$ ) ;
4.    Dsat := Tableau (Taille :  $n$ ) (Défaut :  $0$ ) ;
5.    # Initialisation #
6.    Pour  $i$  de  $0$  à  $n - 1$  faire
7.        Deg.( $i$ )  $\leftarrow d(i)$  ;
8.        Dsat.( $i$ )  $\leftarrow d(i)$ 
9.    Fin Pour ;
10.   # Boucle principale #
11.   Pour  $i$  de  $0$  à  $n - 1$  faire
12.        $x := 0$  ;
13.       # Recherche du sommet de Dsat max #
14.       Pour  $y$  de  $0$  à  $n - 1$  faire
15.           Si Couleur.( $x$ ) =  $-1$  & Dsat.( $y$ ) > Dsat.( $x$ ) faire
16.                $x := y$ 
```

```

17.           Fin Si
18.       Fin Pour ;
19.       # En cas de conflit, prendre celui de Deg max #
20.       Pour  $y$  de 0 à  $n - 1$  faire
21.           Si  $\text{Couleur.}(x) = -1 \ \& \ \text{Dsat.}(y) = \text{Dsat.}(x) \ \& \ \text{Deg.}(y) > \text{Deg.}(x)$ 
22.                $x := y$ 
23.           Fin Si
24.       Fin Pour ;
25.       # Recherche de la plus petite couleur non utilisée dans  $\mathcal{N}(x)$  #
26.       Libre := Tableau (Taille : ???) (Défaut : vrai);
27.       Pour  $y$  dans  $\mathcal{N}(x)$  faire
28.           Si  $\text{Couleur.}(y) \neq -1$  faire
29.               Libre.(Couleur.( $y$ ))  $\leftarrow$  faux;
30.           Fin Si
31.       Fin Pour ;
32.        $index := 0$ ;
33.       Tant Que Libre.( $index$ ) = faux faire
34.            $index := index + 1$ 
35.       Fin Tant Que ;
36.       # Mettre à jour Dsat pour les voisins de  $x$  #
37.       Pour  $y$  dans  $\mathcal{N}(x)$  faire
38.           Si  $\text{Dsat.}(y) = \text{Deg.}(y)$  faire
39.                $\text{Dsat.}(y) \leftarrow 1$ ;
40.           Si  $index \notin \text{couleur}(\mathcal{N}(y))$  faire
41.                $\text{Dsat.}(y) \leftarrow \text{Dsat.}(y) + 1$ ;
42.           Fin Si
43.       Fin Pour ;
44.       # Affecter la couleur donnée par index à  $x$  #
45.        $\text{Couleur.}(x) \leftarrow index$ 
46.   Fin Pour ;
47.   Renvoyer Couleur

```

Donnons maintenant quelques explications sur le code proposé : les deux boucles (lignes 14 à 24) pour rechercher le sommet à colorer sont là pour des raisons de clarté, on pourrait très bien se ramener à une seule boucle. Quant à la mise à jour de $DSAT(y)$, pour un voisin y de x , et ce avant de colorer x , on distingue trois cas :

- y n'a aucun voisin coloré. Alors $DSAT(y) = d(y)$ par définition. Réciproquement l'égalité $DSAT(y) = d(y)$ n'est vraie que dans deux cas : y n'a aucun voisin coloré, ou y a tous ses voisins colorés de couleurs différentes (auquel cas $|\sigma(\mathcal{N}(y))| = |\mathcal{N}(y)| = d(y)$), mais ce n'est pas le cas ici puisque $x \in \mathcal{N}(y)$ et n'est pas encore coloré. En regardant si $DSAT(y) = d(y)$, on est donc sûr qu'il faut régler $DSAT(y)$ à 1.

- y a des voisins colorés, mais aucun de la couleur index : il faut alors incrémenter $DSAT(y)$.

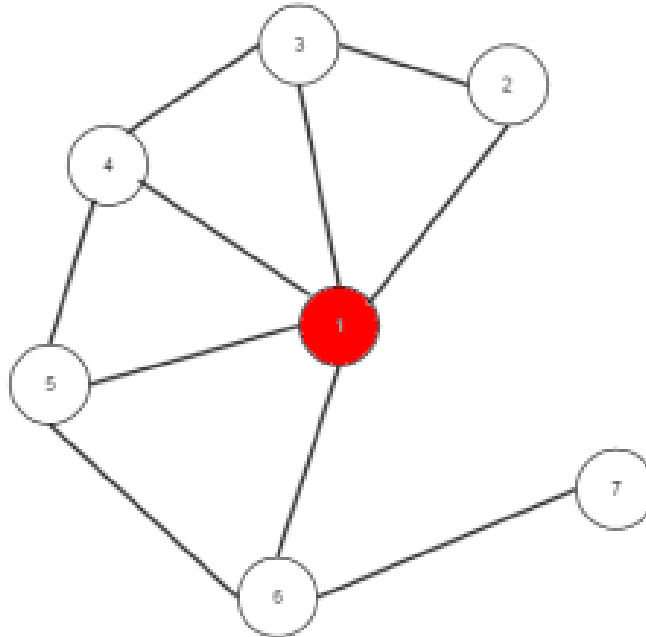
- y a des voisins colorés, dont un au moins de la couleur index : il n'y a alors rien à faire.

La complexité de l'algorithme DSATUR est $O(n^2)$.

Exemple

On prend le sommet de plus haut $DSAT$, soit le sommet 1.

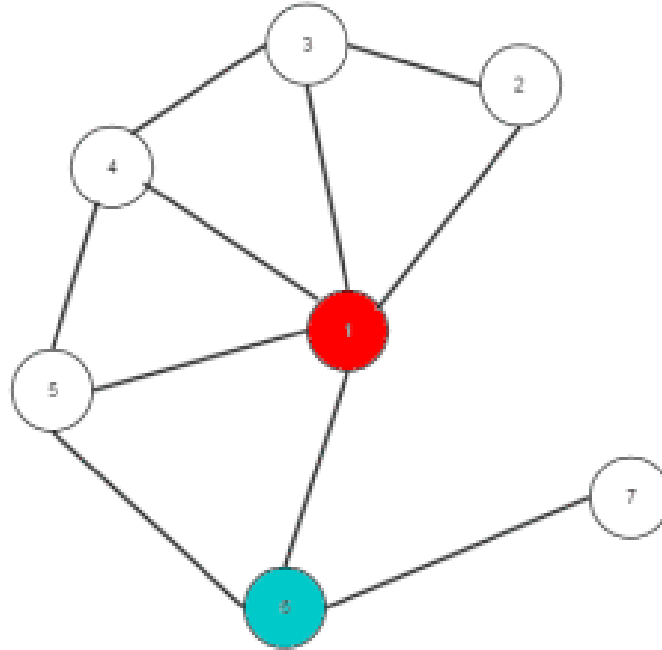
On lui attribue la première couleur minimale, soit le rouge :



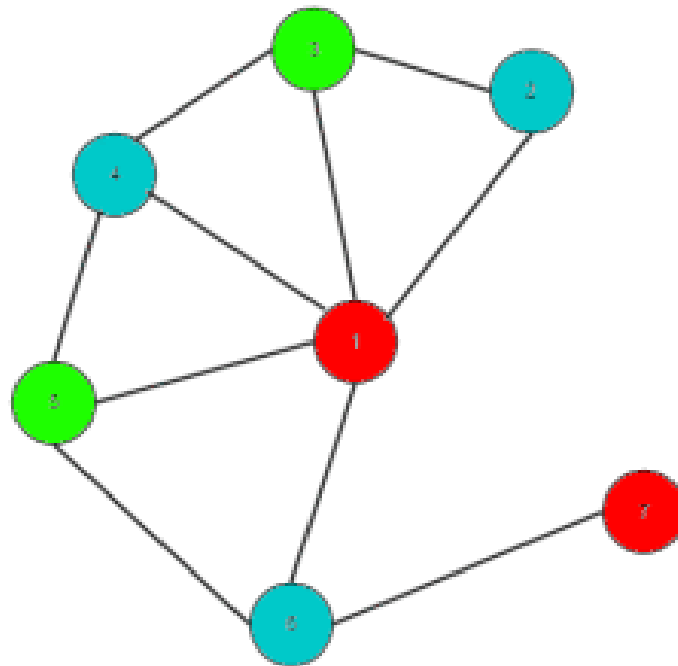
Puis, on passe à la mise à jour des $DSAT$ des voisins du sommet déjà coloré.

On prend le sommet de plus haut $DSAT$, Nous allons prendre le sommet 6.

On lui attribue la deuxième couleur minimale, soit le bleu :



On fait la même chose pour les autres sommets, on obtient le graphe coloré suivant :



3.3 Graphes triangulés : un algorithme linéaire de résolution exacte

Après avoir présenté ces quelques algorithmes de résolution approchés, on s'intéresse maintenant à des algorithmes fournissant une réponse optimale exacte. L'inconvénient des heuristiques précédentes est de ne pas fournir systématiquement de coloration propre optimale. Cependant comme il n'existe pas non plus à l'heure actuelle d'algorithme résolvant le problème général en un temps polynomial, on s'intéresse ici à un cas particulier plus facile : celui des graphes triangulés.

La présence d'un schéma d'élimination dans un graphe triangulé est très utile, car en appelant l'algorithme de coloration séquentielle sur G avec ce schéma d'élimination simplicial, on obtient une coloration propre optimale :

Proposition 9 Soit $G = (V, E)$ graphe et $\sigma : [0, n - 1] \rightarrow V$ schéma d'élimination simplicial. Alors, l'appel :

Coloration séquentielle (Graphe : G) (Numérotation : σ)

Fournit une coloration propre optimale de G .

Démonstration

Il s'agit de montrer que la coloration séquentielle obtenue pour le graphe G est bien une $\omega(G)$ -coloration (étant donné que $\omega(G) = \chi(G)$: le graphe est parfait). En remarquant que $\sigma|_{[0, n-2]}$ est toujours un schéma d'élimination simplicial pour $G' = G[V \setminus \{\sigma(n-1)\}]$, et que G' reste triangulé par hérédité, on raisonne alors par récurrence sur l'ordre du graphe.

- Pour $n = 1$, la propriété est triviale.
- Soit $n > 1$. Supposons que pour tout graphe d'ordre $n - 1$, l'appel Coloration séquentielle avec un schéma d'élimination simplicial fournisse une coloration propre optimale.

On se donne alors G graphe d'ordre n et $\sigma : [0, n - 1] \rightarrow V$ schéma d'élimination simplicial. Notons $x = \sigma(n - 1)$, $G' = G[V \setminus \{x\}]$ ainsi que $c : V \rightarrow \mathbb{N}$ la coloration propre fournie par l'algorithme. Par hypothèse de récurrence, on a donc $c(V \setminus \{x\}) = [0, \omega(G') - 1]$. Deux cas se présentent alors :

★ 1^{er} cas : $\omega(G) = \omega(G') + 1$

Alors il est clair que $|\mathcal{N}(x)| = \omega(G')$, et que tous les voisins de x sont colorés de couleurs différentes. Ainsi le sommet x va être coloré d'une autre couleur, et on aura donc $c(x) = \omega(G')$.

★ 2^{eme} cas : $\omega(G) = \omega(G')$

Alors on a $|\mathcal{N}(x)| < \omega(G')$, et donc $\exists p \in [0, \omega(G') - 1], p \notin c(\mathcal{N}(x))$. L'algorithme va donc colorer x avec une couleur $q \leq p$.

Mots et Ordre Lexicographique

On considère l'alphabet $A = [1, n]$, ensemble fini non vide (ses éléments sont appelés des lettres), que l'on muni de la relation d'ordre habituelle sur les entiers, notée \leq . Un mot sur l'alphabet A est une suite finie d'éléments de A de la forme $x_0 \dots x_{p-1}$. On note A^* l'ensemble des mots sur A . On note $\varepsilon = ()$ le mot vide. La longueur d'un mot $u \in A^*$ sera notée $|u|$: si $u = x_0 \dots x_{p-1} \in A^*$, alors $|u| = p$; et par convention $|\varepsilon| = 0$.

Si $u = x_0 \dots x_{p-1}$ et $i \in [0, p-1]$, on note $u[i] = x_i$, la i -ième lettre de u . Pour deux mots $u = x_0 \dots x_{p-1}$ et $v = y_0 \dots y_{q-1}$, on note $u.v = uv = x_0 \dots x_{p-1}y_0 \dots y_{q-1}$ la concaténation de u et v . Cette opération est associative et admet ε comme élément neutre (autrement dit, (A^*, \cdot) est un monoïde). Un mot v est préfixe d'un autre mot u s'il existe un mot w tel que $u = vw$. On note $Pre(u)$ l'ensemble des préfixes de u .

À partir de la relation d'ordre stricte déduite de \leq , on définit différentes relations binaires sur A^* :

- Pour deux mots u et v , on note $u \triangleleft v$ s'il existe $i \in \mathbb{N}$ vérifiant $i < \min(|u|, |v|)$, $u[i] < v[i]$ et $\forall j \in [0, i-1], u[j] = v[j]$.
- On note ensuite $u < v$ si $u \in Pre(v)$ ou $u \triangleleft v$.
- Enfin, $u \leq v$ lorsque $u < v$ ou $u = v$.

Proposition 10 *On démontre alors les propriétés suivantes sur \triangleleft , $<$ et \leq :*

- a. \triangleleft est une relation d'ordre stricte, non totale, compatible avec \cdot .
- b. $<$ est une relation d'ordre stricte totale, compatible avec \cdot .
- c. \leq est une relation d'ordre totale, compatible avec \cdot .

Démonstration

a. On vérifie les différents axiomes :

- **Irréflexivité** : $\forall u \in A^*, u \not\triangleleft u$

Soit $u \in A^*$. Supposons par l'absurde $u \triangleleft u$. Alors $\exists i < |u|, u[i] < u[i]$, ce qui est impossible car $<$ est une relation d'ordre stricte (et donc irréflexive).

- **Transitivité** : $\forall u, v, w \in A^*, (u \triangleleft v \text{ et } v \triangleleft w) \Rightarrow (u \triangleleft w)$

Soient $u, v, w \in A^*$ tels que $u \triangleleft v$ et $v \triangleleft w$.

Soit $i_0 < \min(|u|, |v|)$ tel que $u[i_0] < v[i_0]$ et $\forall j < i_0, u[j] = v[j]$.

On pose $i = \min(i_0, i_1)$. On a bien $i < \min(|u|, |w|)$, $u[i] < w[i]$ et $\forall j < i, u[j] = v[j]$.

D'où $u \triangleleft w$

- **Non totale** : $\exists u, v \in A^*, (u \not\triangleleft v \text{ et } u \neq v \text{ et } v \not\triangleleft u)$

On prend par exemple $u = \varepsilon$ et $v = a \in A$

- **Compatible** avec \cdot : $\forall u, v, w \in A^*, u \triangleleft v \Rightarrow (uw \triangleleft vw \text{ et } wu \triangleleft wv)$

Découle trivialement des définitions.

Mieux : on a même $\forall u, v, w, w' \in A^*, u \triangleleft v \Rightarrow uv \triangleleft vw'$

b. De même :

- **Irréflexivité** : $\forall u \in A^*, u \not\prec u$

Soit $u \in A^*$. Supposons par l'absurde $u \prec u$. Alors, d'après l'irréflexivité de \triangleleft , on a $u \in \text{Pre}(u) \setminus \{u\}$, ce qui est impossible.

- **Transitivité** : $\forall u, v, w \in A^*, (u \prec v \text{ et } v \prec w) \Rightarrow (u \prec w)$

Soient $u, v, w \in A^*$ tels que $u \prec v$ et $v \prec w$. Quatre cas se présentent alors :

* 1^{er} cas : $u \in \text{Pre}(v) \setminus \{v\}$ et $v \in \text{Pre}(w) \setminus \{w\}$

Alors on a $v = uv'$ et $w = vw'$, avec $v', w' \in A^* \setminus \{\varepsilon\}$. D'où $w = uv'w'$, avec $v'w' \in A^* \setminus \{\varepsilon\}$.

Ainsi on a $u \in \text{Pre}(w) \setminus \{w\}$ et $u \prec w$.

* 2^{eme} cas : $u \triangleleft v$ et $v \triangleleft w$

Alors, par transitivité de \triangleleft , on a encore $u \prec w$.

* 3^{eme} cas : $u \in \text{Pre}(v) \setminus \{v\}$ et $v \triangleleft w$

On écrit $v = uv'$, avec $v' \in A^* \setminus \{\varepsilon\}$.

Soit aussi $i < \min(|v|, |w|)$ tel que $v[i] < w[i]$ et $\forall j < i, v[j] = w[j]$

Alors, si $i < |u|$, on a bien $u \triangleleft w$. Sinon il vient que $u \in \text{Pre}(w) \setminus \{w\}$.

Finalement on a toujours $u \prec w$.

* 4^{eme} cas :

$u \triangleleft v$ et $v \in \text{Pre}(w) \setminus \{w\}$

Soit $i < \min(|u|, |v|)$ tel que $u[i] < v[i]$ et $\forall j < i, u[j] = v[j]$.

On écrit aussi $w = vw'$, avec $w' \in A^* \setminus \{\varepsilon\}$.

Alors, comme $i < |w|$, on en déduit facilement que $u \triangleleft w$.

- **Totale** : $\forall u, v \in A^*, (u \prec v \text{ ou } u = v \text{ ou } v \prec u)$

Soient $\forall u, v \in A^*$. Plusieurs cas se présentent :

o 1^{er} cas : $u \triangleleft v$ ou $v \triangleleft u$

◦ 2^{ème} cas : $u \not\prec v$ et $v \not\prec u$

Alors, $\forall j < \min(|u|, |v|), u[j] = v[j]$. De là on déduit que :

Si $|u| = |v|, u = v$.

Si $|u| < |v|, u \prec v$.

Si $|v| < |u|, v \prec u$.

- **Compatible** avec \cdot : $\forall u, v, w \in A^*, u \prec v \Rightarrow (uw \prec vw \text{ et } wu \prec wv)$

Vrai si $u \triangleleft v$. C'est encore évident si $u \in \text{Pre}(v) \setminus \{v\}$

c. On vérifie facilement :

- **Réflexivité** : $\forall u \in A^*, u \preceq u$

- **Transitivité** : $\forall u, v, w \in A^*, (u \preceq v \text{ et } v \preceq w) \Rightarrow (u \preceq w)$

Découle immédiatement de la transitivité de \prec

- **Antisymétrie** : $\forall u, v \in A^*, (u \preceq v \text{ et } v \preceq u) \Rightarrow (u = v)$

Découle des propriétés de \prec :

Par contra-posée, la transitivité nous donne que $\forall u, v, w \in A^*, (u \not\prec w) \Rightarrow (u \not\prec v \text{ ou } v \not\prec w)$

Appliquée à $u, v \in A^*$ ((on prend $w = u$), et de l'irréflexivité de \prec , on déduit que $(\prec v \text{ et } v \prec u)$ impossible. De fait, si $u \preceq v$ et $v \preceq u$, il vient que $u = v$.

- **Totale** : $\forall u, v \in A^*, (u \preceq v \text{ ou } v \preceq u)$

- **Compatible** avec \cdot : $\forall u, v, w \in A^*, u \preceq v \Rightarrow (uw \preceq vw \text{ et } wu \preceq wv)$

On se donne alors un graphe $G = (V, E)$, où $V = [1, n]$. On note toujours $A = [1, n]$, ou encore $A = \{a_1, \dots, a_n\}$, avec $a_i = i$ pour $1 \leq i \leq n$. On associe alors à chaque sommet $x \in V$ une étiquette, notée $L(x) \in A^*$, un mot sur A , initialement égal à ε . On va donc parcourir les sommets du graphe en effectuant les opérations suivantes :

Itération $i \in [0, n - 1]$

- Prendre un sommet x d'étiquette maximale pour \preceq parmi les sommets non visités.
- Pour chaque voisin $y \in \mathcal{N}(x)$, effectuer la mise à jour $L(y) \leftarrow L(y).a_{n-i}$

Cela nous donne l'algorithme suivant, dit de parcours en largeur lexicographique (ou LexBFS, pour "Lexicographic Breadth First Search") :

Algorithme 4 Parcours en Largeur Lexicographique

Entrée - Un graphe quelconque $G = (V, E)$

Sortie - Une bijection propre $\sigma : V \rightarrow [0, n - 1]$

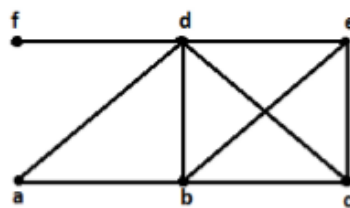
1. $n := |V|;$
 2. $\text{Numero} := \text{Tableau (Taille : } n \text{) (Défaut : } -1 \text{);}$
 3. $\text{Label} := \text{Tableau (Taille : } n \text{) (Défaut : "");}$
 4. **Pour** i de 0 à $n - 1$ **faire**
 5. Soit x tel que $\text{Label}.(x) = \max\{\text{Label}.(y), \text{Numero}.(y) = -1\};$
 6. $\text{Numero}.(x) \leftarrow i;$
 7. **Pour** y dans $\mathcal{N}(x)$ **faire**
 8. **Si** $\text{Numero}.(y) = -1$ **faire**
 9. $\text{Label}.(y) \leftarrow \text{Label}.(x) \wedge \text{String_of_Int}(n - i)$
 10. **Fin Si**
 11. **Fin Pour**
 12. **Fin Pour ;**
 13. Renvoyer Numero
-

Nous obtenons alors un ordre $\sigma : V \rightarrow [0, n - 1]$ sur les sommets du graphe, appelé "ordre lexicographique".

Remarque

Un mot sur A est représenté par une chaîne de caractère. Le mot vide ε est représenté par la chaîne vide "", et la concaténation \cdot est notée \wedge .

Exemple de l'algorithme de Parcours en Largeur Lexicographique



σ	6	5	4	3	2	1
a	\emptyset	$\{6\}$	$\{6, 5\}$	$\{6, 5\}$	★ $\{6, 5\}$	
b	\emptyset	★ $\{6\}$				
c	\emptyset	$\{6\}$	★ $\{6, 5\}$			
d	★ \emptyset					
e	\emptyset	$\{6\}$	$\{6, 5\}$	★ $\{6, 5, 4\}$		
f	\emptyset	$\{6\}$	$\{6\}$	$\{6\}$	$\{6\}$	★ $\{6\}$

TABLE 3.2 – Tableau de Parcours en Largeur Lexicographique

L'ordre σ obtenu est : f, a, e, c, b, d . Nous pouvons vérifier facilement que l'ordre σ est un ordre d'élimination simplicial de G .

3.4 Conclusion

Les quatre algorithmes que nous avons présenté dans le troisième chapitre fournissent des colorations qui diffèrent selon que la coloration obtenue est optimale ou non.

APPLICATIONS ET RÉSULTATS EXPÉRIMENTAUX

4.1 Introduction

Maintenant que nous avons terminé avec l'aspect théorique de la coloration des graphes, et mis en évidence les différents algorithmes permettant de les colorer, il reste à aborder la partie pratique : l'implémentation des algorithmes en C et quelques applications de la coloration des graphes dans le monde réel, ainsi qu'une étude comparative des résultats expérimentaux obtenus avec les divers algorithmes appliqués à ces applications réelles.

4.2 Allocation des fréquences (Gestion du spectre de fréquence et implémentation des réseaux de télécommunications)

4.2.1 Introduction

Depuis les années 80, le problème d'affectation de fréquences a fait l'objet d'études menées par différents chercheurs dans le but d'une meilleure gestion. Le spectre de fréquences qui est attribué aux opérateurs de téléphonie est divisé en canaux fréquentiels. L'allocation de fréquences regroupe les mécanismes et procédures mis en œuvre afin de gérer l'attribution des canaux de fréquences aux demandes de communication. Cette gestion permet de déterminer la qualité du réseau.

Le spectre radioélectrique est devenu une ressource extrêmement précieuse, en raison de son exploitation massive par des systèmes de communication de toutes sortes.

Il faut donc des méthodes novatrices pour gérer le spectre de manière dynamique afin que cette ressource puisse être disponible pour les nouveaux services à même d'éviter les brouillages de signaux.

L'objet de cet application sera de poser la problématique de l'affectation de fréquences et la traduire en théorie des graphes ; Ceci nous permettra d'une part de proposer des solutions à même de résoudre cette problématique et d'autre part, de présenter quelques définitions de base qui nous permettront de mieux comprendre la problématique posée.

Le but de cette étude sera aussi de satisfaire la demande de chaque station de base en terme de fréquences tout en minimisant l'ensemble des interférences ainsi que la partie du spectre radio.

Pour cela, et à l'issu de ce travail nous proposons l'implémentation de quelques algorithmes de coloration des graphes permettant la gestion du spectre de fréquence sur les stations de bases en utilisant un minimum de fréquences pour une couverture maximale.

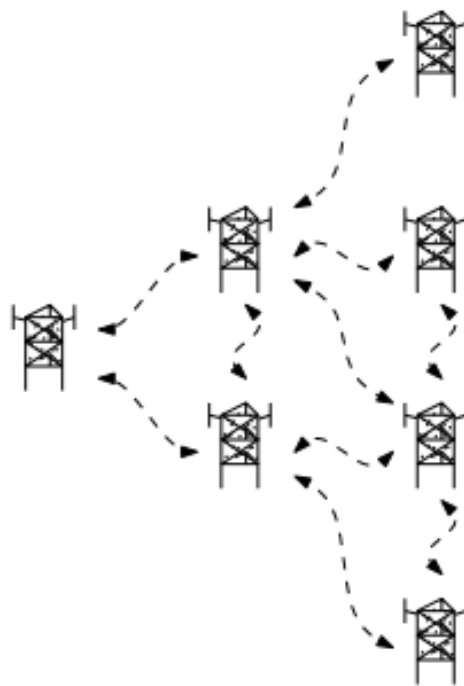


FIGURE 4.1 – Un réseau d'antennes

4.2.2 Quelques définitions

L'attribution des fréquences : est l'ensemble des mécanismes qui permettent de dé-

finir comment sont réparties les fréquences entre les différents acteurs. L'enjeu principal est d'éviter les interférences entre les émetteurs.

Fréquence : En physique, la fréquence désigne en général la mesure du nombre de fois qu'un phénomène périodique se reproduit par unité de temps ; C'est-à-dire le nombre de fois qu'un phénomène temporel régulier se reproduit identique à lui-même par intervalle de temps donné. L'unité de fréquence étant l'hertz (hz).

Spectre de fréquence : Le spectre de fréquence est la partie du spectre électromagnétique qui achemine les ondes radio ; Il est subdivisé en neuf bandes de fréquences, désignées par des nombres entiers consécutifs.

4.2.3 Les objectifs de la gestion du spectre

- Garantir une plus grande facilité d'utilisation du spectre.
- Rationaliser l'usage du spectre, même dans un environnement où la fréquence n'est pas encore une ressource rare.
- Garantir la disponibilité des fréquences.
- Répondre au besoin de développement des télécommunications et des radiocommunications nationales.
- Répondre aux besoins de la sécurité et de la défense nationale.
- Contrôler l'utilisation (conformité) du spectre.

La non gestion rationnelle du spectre de fréquence entraine automatiquement une mauvaise qualité du signal par la création d'interférences.

4.2.4 Les interférences

On parle d'interférence lorsqu'un point donné de l'espace de couverture reçoit en plus du signal utile (assurant le service) un signal dit interférent de puissance relativement élevée et porté sur une fréquence identique ou adjacente.

Il existe deux grands types d'interférences, mais dans notre cas nous intéressons à celles qui sont dues à la réutilisation d'une même fréquence (interférence Co-canal).

4.2.5 Problématique

Aujourd'hui les opérateurs radio doivent faire face à un double déficit : répondre à une augmentation croissante du trafic tout en maintenant une bonne qualité radio. La difficulté du problème résulte du fait qu'une solution acceptable doit satisfaire des contraintes multiples, la plupart des fois contradictoires.

L'une des contraintes les plus sévères c'est le nombre restreint de fréquences (canaux) disponibles pour l'allocation (le spectre radio étant une ressource très limitée). Cette contrainte impose un degré élevé de réutilisation des fréquences, fait qui augmente la probabilité d'interférence.

Les gestionnaires disposent d'un nombre limité de fréquences pour couvrir la totalité de la zone d'étude (un territoire par exemple). Pour cela, la réutilisation des fréquences est par conséquent inévitable pour augmenter la capacité du réseau et répondre à la demande de plus en plus importante. Une distribution optimale du spectre de fréquences disponibles sur les stations doit garantir un écoulement maximal du trafic tout en minimisant les interférences. L'écoulement de trafic porté par ces zones est alors conditionné par la qualité de l'affectation des fréquences.

Afin de satisfaire cette demande, il est indispensable d'exploiter cette ressource d'une manière rationnelle et optimale ; Il est donc légitime de s'intéresser à une modélisation plus fine de la problématique de planification de fréquences. Un tel projet nécessite l'utilisation de méthodes scientifiques basées sur des techniques de recherche opérationnelle.

4.2.6 Modélisation mathématique du problème par la théorie des graphes

Comme mentionné précédemment, les gestionnaires sont souvent confrontés à de multiples problèmes et leur résolution s'avère souvent une tâche difficile. Ces problèmes se présentent sous forme de données, de contraintes dont on doit en tenir compte.

Pour arriver à résoudre un problème donné, nous devons commencer par interpréter tous ses paramètres et les transformer sous des formes qu'on peut gérer. La première étape dans la résolution d'un problème consiste en sa projection dans un espace permettant ainsi diverses manipulations sur le problème projeté. Ce dernier s'appelle le modèle associé au problème.

La modélisation est donc une traduction des paramètres du problème dans un langage

accessible par la méthode de résolution utilisée, ou bien c'est une façon de décrire le problème sous une forme qui introduit sa réalisation.

Enfin, la modélisation d'un problème doit pouvoir donner une interprétation aux solutions concrètes répondant aux besoins du problème réellement posé.

Les données du problème :

N : nombre de stations.

N_F : nombre de fréquences disponibles.

$d(n_i, n_j)$: la distance entre deux stations n_i et n_j .

Contrainte :

Respecter la contrainte d'interférence, ie : éviter d'assigner la même fréquence à deux stations voisines.

Objectifs :

Trouver un bon plan de fréquence qui doit minimiser l'ensemble des interférences.

Soit un graphe $G = (X, E)$ défini par :

X : L'ensemble des sommets du graphe représentent les stations.

E : L'ensemble des arêtes du graphe représentent les risques d'interférences, Il existe une arête $[x_i, x_j]$ de E ssi x_i est voisin avec x_j .

Pour résoudre le problème d'affectation de fréquences, on utilise la coloration des sommets qui consiste à affecter à tous les sommets du graphe une couleur (fréquence) de façon que chaque paire de sommets adjacents soit de couleurs différentes ; En d'autre termes, s'il existe une arête $[x_i, x_j]$ de E alors on a $c(x_i) \neq c(x_j)$.

Le nombre minimum de couleurs nécessaires pour colorier ce graphe en respectant cette contrainte, est appelé le nombre chromatique $\chi(G)$.

L'application de la théorie des graphes va nous permettre de trouver le nombre minimal de fréquences allouées aux stations de bases et qui minimise l'intégralité des interférences.

4.2.7 Problèmes tests

Afin de valider notre approche, nous exploitons les données générées aléatoirement présentées ci dessous à savoir.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	-	45	50	80	41	75	63	50	93	30	88	42	90	78	55	87	99	72	50	49	63	88	94
1	45	-	39	88	97	56	87	91	98	94	50	80	44	70	44	38	50	42	77	91	85	82	69
2	50	39	-	42	33	42	94	94	69	83	82	95	50	43	71	96	82	84	58	91	78	89	90
3	80	88	42	-	50	41	36	98	50	93	72	80	95	45	68	68	89	90	75	71	78	90	84
4	41	97	33	50	-	75	84	35	41	72	55	95	84	99	86	92	77	99	82	73	98	59	66
5	75	56	42	41	75	-	50	85	94	75	73	99	94	47	82	75	85	67	82	94	72	58	84
6	63	87	94	36	84	50	-	86	42	94	77	82	59	69	71	74	78	75	99	84	75	49	82
7	50	91	94	98	35	85	86	-	31	46	99	84	75	68	57	85	74	65	74	58	48	75	82
8	93	98	69	50	41	94	42	31	-	99	83	75	69	63	88	74	71	86	76	95	72	50	81
9	30	94	83	93	72	75	94	46	99	-	80	86	74	63	84	72	58	91	44	75	50	77	81
10	88	50	82	72	55	73	77	99	83	80	-	38	72	81	99	29	84	71	92	66	74	63	50
11	42	80	95	80	95	99	82	84	75	86	38	-	85	90	60	78	68	74	76	26	85	75	45
12	90	44	50	95	84	94	59	75	69	74	72	85	-	80	50	76	66	84	65	98	68	70	73
13	78	70	43	45	99	47	69	68	63	63	81	90	80	-	84	99	63	63	80	80	97	90	61
14	55	44	71	68	86	82	71	57	88	84	99	60	50	84	-	58	70	44	70	82	63	75	74
15	87	38	96	68	92	75	74	85	74	72	29	78	76	99	58	-	50	74	71	84	99	83	21
16	99	50	82	89	77	85	78	74	71	58	84	68	66	63	70	50	-	50	66	73	81	85	92
17	72	42	84	90	99	67	75	65	86	91	71	74	84	63	44	74	50	-	73	66	85	92	88
18	50	77	58	75	82	82	99	74	76	44	92	76	65	80	70	71	66	73	-	44	73	71	69
19	49	91	91	71	73	94	84	58	95	75	66	26	98	80	82	84	73	66	44	-	85	72	65
20	63	85	78	78	98	72	75	48	72	50	74	85	68	97	63	99	81	85	73	85	-	76	84
21	88	82	89	90	59	58	49	75	50	77	63	75	70	90	75	83	85	92	71	72	76	-	99
22	94	69	90	84	66	84	82	82	81	81	50	45	73	61	74	21	92	88	69	65	84	99	-

TABLE 4.1 – Tableau des distances entre les stations

Nous avons défini 23 stations et les distances entre elles, on désire attribuer des canaux de fréquences à ces stations. Deux stations distantes de moins de 50 km ne peuvent pas avoir le même canal afin d'éviter les interférences. Le but est de chercher le nombre minimum

de canaux distincts nécessaires pour éviter toute interférence.

Afin de convertir le problème d'allocation des fréquences en un problème de coloration, On va traduire le tableau ci-dessus en terme de graphe dont les sommets sont les stations, et les arêtes relient deux stations qui interfèrent entre elles.

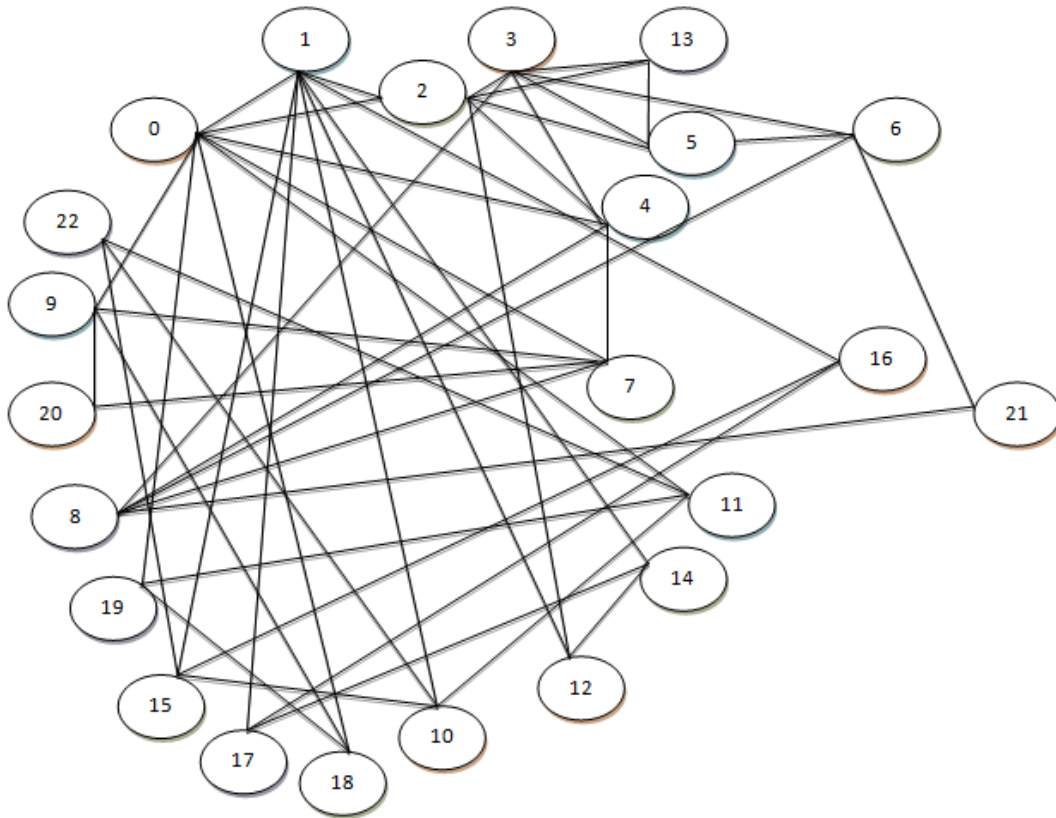


FIGURE 4.2 – Graphe d'interférences correspondant au tableau ci-dessus, dont les sommets sont les stations, et les arêtes relient deux stations qui interfèrent entre elles

4.2.8 Résultats expérimentaux

Après la modélisation du réseau des stations par le graphe appelé graphe d'interférences. On va maintenant s'intéresser à sa coloration par les différents algorithmes détaillés précédemment, les fréquences à allouer correspondent à des couleurs avec lesquelles nous colorons les sommets du graphe. La coloration doit être une coloration propre : deux sommets adjacents ne peuvent recevoir la même couleur, en plus, puisque le graphe est non-orienté, la relation "interfère" est symétrique.

Les algorithmes ont été implémentés de plusieurs manières en C : usage de différents tris

pour Welsh and Powell (tri par sélection, tri par fusion, tri par dénombrement), différentes structures de données pour DSATUR, etc.

Les colorations obtenues par les divers algorithmes sont les suivantes :

Soit les couleurs hiérarchisées :



• L'algorithme de coloration séquentielle :

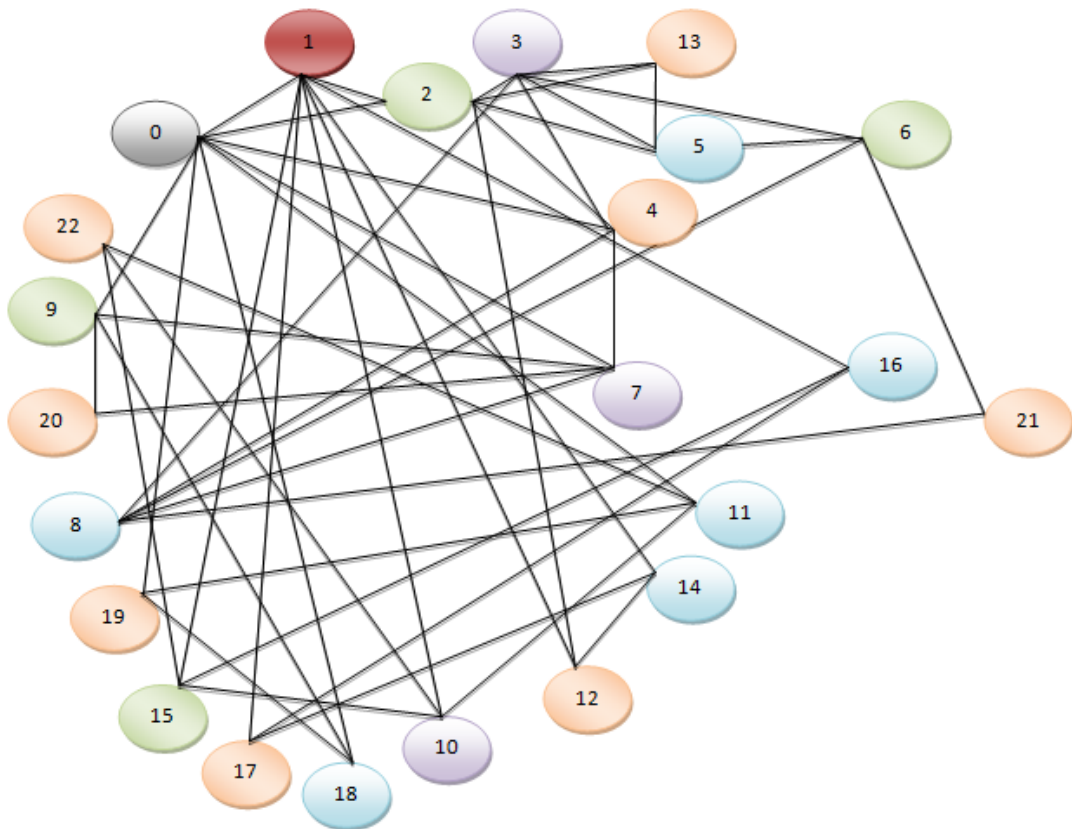


FIGURE 4.3 – Coloration obtenue par l'algorithme de coloration séquentielle

Pour cet algorithme, on choisit un ordre quelconque sur les sommets du graphe. Dans notre cas, on a parcouru les sommets du graphe en commençant par le sommet de plus grand indice jusqu'à le sommet de plus petit indice. On a obtenu une coloration avec 6 couleurs et 0.47s comme temps d'exécution du programme en C.

- L'algorithme de Welsh and Powell :

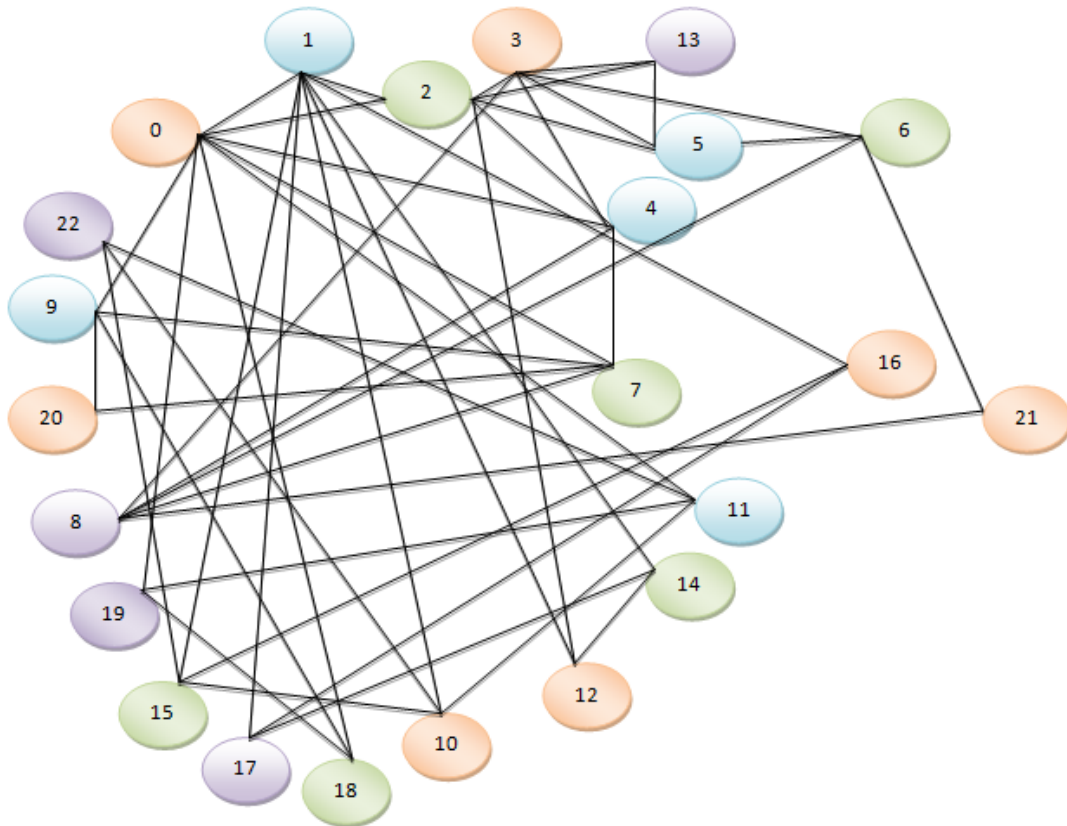


FIGURE 4.4 – Coloration obtenue par l'algorithme de Welsh and Powell

Cette fois ci, avec l'algorithme de Welsh and Powell, on a parcouru la liste des sommets dans l'ordre décroissant. Le nombre de couleurs obtenu est 4 couleurs. Le tri par fusion utilisé s'avère plus efficace (le temps d'exécution du programme en C est 0.72s) sur ce graphe que le tri par dénombrement (le temps d'exécution du programme en C est 0.93s).

- L'algorithme de DSATUR :

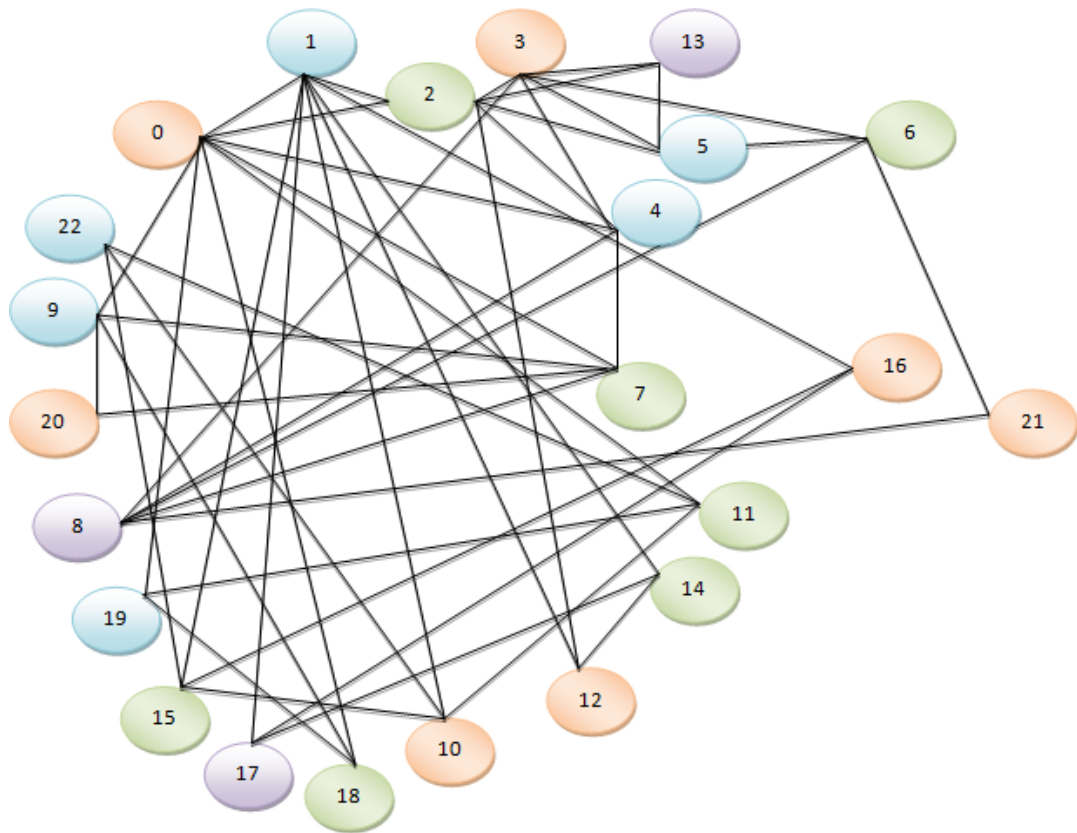


FIGURE 4.5 – Coloration obtenue par l'algorithme de DSATUR

L'algorithme de DSATUR a donné une coloration avec 4 couleurs et un temps d'exécution 1.18s.

• L'algorithme de LexBFS :

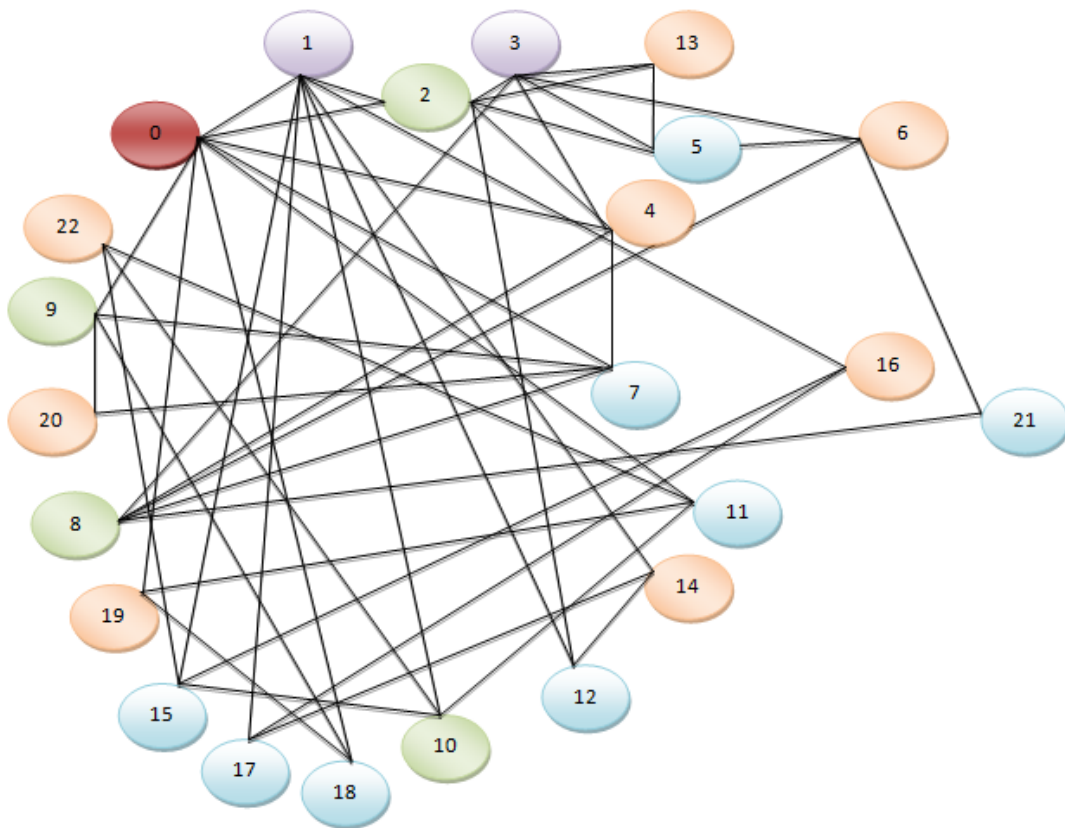


FIGURE 4.6 – Coloration obtenue par l'algorithme de LexBFS

L'algorithme de LexBFS a donné une coloration avec 5 couleurs et un temps d'exécution 0.33s.

4.2.9 Comparaison des résultats

On va voir la différence d'efficacité en terme de nombre de couleurs utilisées et de temps d'exécution par les différents algorithmes sur le graphe d'interférences.

Comme on a déjà vu, DSATUR et Welsh and Powell sont plus efficaces en terme de nombre de couleurs utilisées que la coloration séquentielle et LexBFS, mais en ce qui concerne le temps d'exécution LexBFS et la coloration séquentielle sont plus efficace que DSATUR et Welsh and Powell.

Le nombre de couleurs obtenu par les deux algorithmes DSATUR et Welsh and Powell est le même, ils fournissent tous les deux une bonne coloration par rapport à l'algorithme

de coloration séquentielle et LexBFS.

On remarque aussi que DSATUR est moins rapide que Welsh and Powell, ce dernier est deux fois plus efficace si on utilise le tri par fusion que l'algorithme de tri par dénombrement. LexBFS est plus rapide que la coloration séquentielle, ce dernier est plus rapide que Welsh and Powell.

4.3 Conclusion

Dans cette application, nous nous sommes intéressés au problème de la gestion rationnelle du spectre de fréquences par des méthodes propres à la recherche opérationnelle, nous avons procédé à cette étude en utilisant quatre algorithmes de coloration, trois algorithmes pour des graphes quelconques et un autre algorithme qui s'applique sur les graphes triangulés.

Après avoir implémenté ces différents algorithmes en C, nous les avons testé selon les mêmes contraintes à notre modèle de réseau pour la recherche d'un plan de gestion de fréquences et évaluer les performances de chaque algorithme.

Les résultats obtenus ont été satisfaisants notamment celles obtenus par les deux heuristiques.

A l'issue de ce travail, nous pensons que ces algorithmes de coloration seront d'un apport certain pour l'amélioration de la gestion du spectre de fréquences.

Conclusion et perspectives

Dans ce mémoire, nous avons abordé le problème de coloration des graphes qui est un champ majeur et très actif de la théorie des graphes.

Pour commencer, nous avons rappelé les notions de base de la théorie de la complexité et de théorie des graphes.

Dans un second temps, nous nous sommes intéressés à la coloration des sommets et des arêtes et leurs propriétés.

Les quatre algorithmes que nous avons présenté dans le troisième chapitre fournissent des colorations qui diffèrent selon que la coloration obtenue est optimale ou non.

Dans le quatrième chapitre, nous avons abordé une application de la coloration des graphes dans le monde réel et nous avons appliqué les différents algorithmes implémentés en C à cette application. Nous concluons que les différences d'efficacité constatées, ne dépendent pas seulement des algorithmes considérées, mais aussi de l'implémentation effectuée. La coloration de graphe est un domaine de recherche très actif et, comme nous avons pu le voir, ses applications sont multiples.

On notera également que le problème de coloration simple abordé n'est pas exhaustif, et qu'il peut s'étendre de multiples façons : coloration avec contraintes (nombre limité de couleurs, sommets avec couleurs imposées, etc.), multicoloration (allocation d'une liste de couleurs à chaque sommet), etc

En termes de perspectives, nous visons à développer d'autres algorithmes qui seront plus efficaces et plus performants que ceux qui ont été présenté précédemment, et de chercher des algorithmes fournissant des solutions optimales exactes pour des cas particuliers de graphe. Nous souhaitons aussi intégrer d'autres contraintes liées aux problèmes étudiés.

Bibliographie

- [1] Aloïs Brunel, Les Graphes Parfaits, École Normale Supérieure de Lyon, 4 avril 2009.
- [2] A. Ghouila-Houri, Sur une conjecture de Berge (mimeo), 1960.
- [3] B. Lévêque, Coloration de graphes :structures et algorithmes, l'Université Joseph Fourier, 15 Octobre 2007.
- [4] C. Berge, Les problèmes de coloration en théorie des graphes, Publications de l'Institut de Statistiques de l'Université de Paris (1960) 123-160.
- [5] C. Berge, Some classes of perfect graphs, in Six papers on Graph Theory, Calcutta, 1963, Indian Statistical Institute.
- [6] C. H. Papadimitriou. Computational complexity. Addison-Wesley, Reading, 1994.
- [7] C.Mezioud, Recherche sur la Résolution des Problèmes Complexes d'Affectation de Fréquences Basses Bandes pour les Opérateurs de la Téléphonie Mobile, Université Mentouri de Constantine - Faculté des sciences de l'ingénieur, Département d'Informatique, thèse pour l'obtention de grade de docteur en science soutenue publiquement le 26/05/2011.
- [8] D.R. Fulkerson, Anti-blocking polyhedra, Journal of Combinatorial Theory B 12 (1972) 50-71.
- [9] Euclide, Les éléments, Volume II, Livres V à IX, Traduction du texte de Heiberg et commentaires par Bernard Vitrac, Presses universitaires de France, Paris, 1994.
- [10] J. Bang-Jensen and G. Gutin. Alternating cycles and paths in edge-coloured multigraphs : a survey, Discrete Mathematics, 165/166, 39-60, 1997.

- [11] J. Edmonds. Minimum partition of a matroid into independent subsets. J. Res. Nat. Bur. Standards Sect. B, 69 :67–72, 1965.
 - [12] J. Hao, R. Dorne, P. Galinier "Tabu Search for Frequency Assignment in Mobile Radio Networks" Journal of heuristics, N 4, 1998, pp 47-62.
 - [13] L. Lovász, A characterization of perfect graphs, Journal of Combinatorial Theory B 13 (1972) 95-98.
 - [14] L. Lovász, Normal hypergraphs and the perfect graph conjecture, Discrete Mathematics 2 (1972) 253-267.
 - [15] M. Chudnovsky, G. Cornuéols, X. Liu, P. Seymour, K. Vušković, Recognizing Berge Graphs, Combinatorica 25 (2005) 143-186.
 - [16] M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas, The strong perfect graph theorem, Annals of Mathematics 164 (2006) 51-229.
 - [17] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (1981) 169-197.
 - [18] Michael R. Garey and David S. Johnson, computers and interactability a Guide to the Theory of NP-Completeness.
 - [19] N. Trotignon, Graphes parfaits : Structure et algorithmes, l'Université Grenoble I, Joseph Fourier, 28 septembre 2004.
 - [20] P. Gambette. Cours d'algorithmique des graphes du MPRI, 18, 9 janvier 2008.
 - [21] P. Hansen, A. Hertz et J. Kuplinsky, Bounded Vertex Colorings of Graphs. Discrete Math. 111,1993, 305-312
 - [22] R. A. Brualdi and J. J. Q. Massey. Incidence and strong edge colorings of graphs. Discrete Math., 122(1-3) :51–58, 1993.
 - [23] R. Borndorfer, A. Eisenblatter, M. Grotschel, A. Martin, Frequency Assignment in Cellular Phone Networks. ZIB-Repport, 97-35,1997.
 - [24] R. Diestel, Graph Theory, Springer, New York, second édition, 2000.
 - [25] R. Karp, Reducibility among combinatorial problems, Complexity of Computer Computations 85-103, Plenum Press, 1972.
-