

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département Recherche opérationnelle

MÉMOIRE DE MASTER RECHERCHE

En

Recherche opérationnelle

Option

Modélisation Mathématique et Évaluation de Performances dans les Réseaux

Thème

Problèmes de colorations et Applications

Présenté par :

SAMAH lynda
ZIZI Rebiha

Soutenu le 3 Juillet 2019 devant le jury composé de :

Président	Dr O.LEKADIR	U. A/Mira Béjaïa.
Rapporteur	Dr k. KABYL	U. A/Mira Béjaïa.
Examineur	Dr S. TAOUINET	U. A/Mira Béjaïa.
Examineur	Dr M. SOUFIT	U. A/Mira Béjaïa.

** Remerciements **

Louange A Dieu, le miséricordieux, sans Lui rien de tout cela n'aurait pu être.

Au terme de ce travail nous exprimons nos plus sincères remerciements à notre promoteur **Mr KABYL.K** pour l'honneur qu'il nous a fait en acceptant de nous encadrer. Ces conseils précieux ont permis une bonne orientation dans la réalisation de ce modeste travail.

Nous tenons également à remercier les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger ce travail, et d'avoir consacré leurs temps pour sa lecture.

Nous tenons à adresser nos vifs remerciement à **Mr TOUATI.S** qui n'a ménagé aucun effort pour répondre à nos questions et satisfaire notre curiosité.

Nous tenons à adresser nos vifs remerciement à nos enseignants pour le savoir qui nous ont transmis durant notre cursus.

Enfin nous tenons à rendre hommage à toutes nos famille et nos amis pour le soutien qu'ils nous ont apportés durant toutes ces années d'études.

※ Dédicaces ※

Je Dédie ce modeste travail à :

A celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur. et ma réussite, à ma mère que dieu la garde et la protège.

A mon père pour ces sacrifices et
les efforts commis jour et nuit pour mon éducation et formation
toutes mes joies, mon amour et ma reconnaissance à mon âme Ali.

A Més très précieuse soeurs : Djahida et Nadjima,

les mots ne peuvent résumer
ma reconnaissance et mon amour à votre égard.

A Més chers frères : Hakim, Nacer, Sofiane et Aissa que j'aime tant.

A Més oncles et Tantes.

A tout les membres de la famille SAMAH.

A tous mes amis avec lesquels j'ai partagé mes moments de joie et de bonheur.

A tous ceux qui m'ont soutenu de près ou de loin.

A toute la promotion R.O 2018-2019.

M. SAMAH LYNDA

※ Dédicaces ※

Tout simplement je dédie ce travail :

A mes chers parents

Aucune dédicace ne saurait être assez éloquente pour exprimer mon respect, mon amour et ma considération pour les sacrifices que vous avez consenti pour moi

A mes chers et adorables frères et soeur

Dyhia, la prunelle de mes yeux, tu représente le symbole de bonté par excellence.

Messi et Yani, les perles de la famille, que dieu, vous protèges.

A ma cher cousine Garti Warda

A la famille ZIZI et Garti.

A mes amies

A une spéciale personne dans ma vie, à celui que je porte beaucoup de tendresse et de respect, à toi karim.

A tous ceux qui ont participé, de près ou de loin à la réalisation de ce travail.

A toute la promotion R.O 2019.

M. ZIZI REBIHA

Table des matières

Table des matières	i
Table des figures	iv
Introduction générale	1
1 Quelques notions élémentaires de la théorie des graphes	3
1.1 Définitions et concepts de base	3
1.1.1 Graphe	3
1.1.2 Sous graphe	7
1.1.3 Graphe partiel	7
1.1.4 Sous graphe partiel	7
1.1.5 Clique	7
1.1.6 Stable	9
1.1.7 Couplage	9
1.1.8 Transversal	10
1.2 Chaîne et Cycle dans les graphes	10
1.3 Connexité dans les graphes	11
1.3.1 Isthme	11
1.4 Représentation des graphes	12
1.5 Opérations classiques sur les graphes	13
1.5.1 Somme Cartésienne de deux graphes	14
1.5.2 Produit Cartésien de deux graphes	14
1.5.3 Subdivisions de graphes	15
1.5.4 Morphismes de graphes	16
1.5.5 Isomorphismes	16
1.6 Distances et Intervalles dans les graphes	17
1.6.1 Distances dans les graphes	17
1.6.2 Intervalles	18
1.7 Quelques graphes particuliers	18
1.7.1 Graphe simple	18

1.7.2	Graphe multiple	18
1.7.3	Graphe K-régulier	18
1.7.4	Graphe complet	19
1.7.5	P-graphe	19
1.7.6	Graphe biparti	19
1.7.7	Graphe biparti complet	20
1.7.8	Graphe biparti équilibré	20
1.7.9	Graphe planaire	20
1.7.10	Arbres	20
1.7.11	Étoile	22
1.7.12	Chenille	22
1.7.13	Graphe valué (pondéré)	22
1.7.14	Graphe de hamming	22
1.7.15	Graphe de Petersen	23
1.7.16	Hypercube Q_n	23
1.7.17	La Roue	24
1.8	Conclusion	25
2	Complexité algorithmique	26
2.1	Introduction	26
2.2	Définitions	26
2.2.1	Algorithme	26
2.2.2	Algorithmique	26
2.3	Complexité d'un algorithme	27
2.4	Calcul de complexité	28
2.4.1	Le temps d'exécution	28
2.4.2	Règle de calcul simplificatif	29
2.5	Différents types de complexité	29
2.6	Comment mesurer l'efficacité d'un algorithme	29
2.7	La notion de problème	30
2.7.1	Un problème en général	30
2.7.2	Problèmes de décision et d'optimisation	30
2.8	Classes des problèmes	31
2.8.1	Les classes P (Polynomial time) et NP (Non deterministic Polynomial time)	32
2.8.2	La classe NP-complet	33
2.9	Conclusion	34
3	Coloration dans les graphes	35
3.1	Introduction	35
3.2	Définitions	35

3.2.1	Coloration des sommets	35
3.2.2	Coloration des arêtes	36
3.2.3	Nombre chromatique	37
3.3	Types de coloration	37
3.3.1	Coloration simple	37
3.3.2	Coloration K-équitable	37
3.3.3	Coloration avec Satisfaction de Contraintes	42
3.3.4	Coloration d'arêtes ℓ -distance	44
3.3.5	Coloration par sommes sur les arêtes	46
3.3.6	Coloration par multi-ensembles	47
3.3.7	Coloration par multi-sommes	50
3.4	Les méthodes de coloration simple des graphes	57
3.4.1	Algorithme de coloration de WELSH et POWELL [6]	57
3.4.2	Algorithme DSATUR [25]	60
3.4.3	Algorithme de Coloration à jeton [11]	62
4	Nouvelle classe de graphe coloriable par sommes et Applications	64
4.1	Introduction	64
4.2	Coloration par sommes sur les arêtes d'une classe de graphe	64
4.3	Introduction de CodeBlocks	70
4.4	Résolution du problème d'allocation de fréquences	71
4.4.1	Concept cellulaire	71
4.4.2	Modélisation par la théorie des graphes	73
4.4.3	Résolution de problème d'allocation de fréquences	74
4.5	Application de la coloration de graphe pour la carte géographique des daïras de la wilaya de Annaba	80
4.6	Conclusion	83
	Conclusion et perspectives	84
	Bibliographie	85

Table des figures

1.1	Exemple d'un graphe à 4 sommets et 4 arêtes.	4
1.2	Graphe orienté à 5 sommets et 7 arcs.	4
1.3	Un Graphe non orienté à 4 sommets et 3 arêtes.	5
1.4	Graphe à 8 sommets et 9 arêtes.	6
1.5	Les graphe iduite à partir de graphe G	8
1.6	9
1.7	Couplage parfait.	10
1.8	Graphe non connexe (Sous graphe connexe)	11
1.9	Isthme.	12
1.10	Un Graphe G.	13
1.11	Somme Cartésienne de $K_2 \square C_4$	14
1.12	Produit cartésien de deux graphes $G_1 \times G_2$	15
1.13	Subdivision de graphes	15
1.14	Homomorphisme de G dans H.	16
1.15	Homomorphisme de G_1 dans G_2 et isomorphisme de G dans H.	17
1.16	Graphe complet K_5	19
1.17	Graphe biparti.	19
1.18	20
1.19	Une chenille	22
1.20	Graphe de Petersen	23
1.21	Q_0, Q_1, Q_2, Q_3	24
1.22	Roue W_{10}	24
3.1	La Coloration des arêtes	36
3.2	Coloration simple.	37
3.3	Coloration 3-équitable.	38
3.4	Exemple 1.	38
3.5	Exemple 2.	39
3.6	Graphe auquel des arêtes ont été ajoutées pour obtenir un graphe connexe dont toutes les faces intérieures sont bornées par un triangle.	41
3.7	Problème de coloriage de carte et son graphe des contraintes associé.	43

3.8	Coloration d'arêtes 1-distance de P_5 .	45
3.9	Exemples de la coloration d'arêtes ℓ -distance de $Q_{i,1 \leq i \leq 4}$. Le sousgraphe $S_{\ell,d}$ est donne par les arêtes en gras. Les couleurs encadrées sont les couleurs des arêtes connecteurs.	46
3.10	Exemple de coloration par sommes sur les arêtes.	47
3.11	Exemple de coloration par multi-ensembles.	48
3.12	Exemple de décomposition en sous-graphes localement irréguliers.	49
3.13	Exemple d'un graphe non décomposable	50
3.14	Exemple de coloration faible par multi-sommes.	52
3.15	Exemple de coloration standard par multi-sommes.	53
3.16	Exemple de coloration forte par multi-sommes.	54
3.17	Deux façons de colorier P 3 avec deux couleurs et un poids.	54
3.18	C_4 est fortement (2, 1)-chromatique	55
3.19	Un Graphe 1.	61
3.20	Un Graphe $K_{2,3}$.	62
4.1	Graphe H_1	64
4.2	Graphe H_2	65
4.3	Coloration par sommes sur les arêtes de H_1	65
4.4	Coloration par sommes sur les arêtes de H_2	66
4.5	Construction de H_3 à partir de deux copies disjointes de H_1	66
4.6	Coloration par sommes de H_5 et H_7	67
4.7	Coloration par sommes de H_4 et H_6	68
4.8	La Décomposable en sous-graphes localement irréguliers de H_6	69
4.9	La Décomposable en sous-graphes localement irréguliers de H_7 .	69
4.10	Modèle du concept cellulaire.	71
4.11		72
4.12	Un réseau d'antennes	73
4.13	Graphe cellulaire	74
4.14	Le graphe associé à Itération 1	75
4.15	Le graphe associé à Itération 2	76
4.16	Le graphe associé à Itération 3	76
4.17	Le graphe correspond à la carte géographique de la wilaya de Annaba.	81
4.18	Implémentation en CodeBlocks	82

Introduction générale

La recherche opérationnelle peut être définie comme étant l'ensemble des méthodes et des techniques rationnelles d'analyse des phénomènes de management du système d'information utilisable pour élaborer de meilleures décisions. Elle propose des modèles conceptuels pour analyser des situations complexes et permet aux décideurs de faire les choix les plus efficaces.

Les domaines et les outils faisant part de la recherche opérationnelle sont très vastes, parmi ces outils, on trouve la théorie des graphes qui est devenue l'un des instruments les plus efficaces pour résoudre la plupart des problèmes discrets que pose la RO. La coloration de graphe est probablement l'un des problèmes d'optimisation combinatoire les plus étudiés en informatique et en mathématiques. Tous les graphes sont coloriables, le problème consiste à colorier le graphe avec un nombre minimum de couleurs, ce problème peut être formulé d'une façon très simple : Colorier les sommets tel qu'il n'existe pas deux sommets adjacents de la même couleur. D'un point de vue complexité théorique, la coloration de graphe a été l'un des premiers problèmes démontrés NP-complet.

La notion de coloration de graphes correspond en effet à la partition d'un ensemble d'objets en plusieurs classes suivant certaines règles. L'ensemble d'objets peut être constitué des sommets (respectivement des arêtes, des arêtes et des sommets ou d'incidences) d'un graphe, les classes sont des ensembles de sommets (arêtes et de sommets ou d'incidences) de même couleur et les règles sont des règles de coloration de sommets.

Différents types de coloration peuvent être définis suivant les règles considérées : coloration propre, arête-coloration, coloration orientée, coloration totale, coloration d'incidence, etc. Par exemple, lorsqu'on donne des couleurs différentes à deux sommets joints par une arête dans le graphe, il s'agit d'une coloration propre du graphe et lorsqu'on colorie deux arêtes incidentes à un sommet dans le graphe par deux couleurs distinctes, il s'agit d'une arête-coloration du graphe.

Il existe de nombreux autres types de coloration de graphes et de nombreux résultats que nous ne détaillerons pas ici mais nous pouvons citer l'excellent ouvrage de Jensen et Toft, "Graph Coloring

Problems”, qui rassemble la grande majorité de ces résultats ainsi que de nombreuses questions ouvertes.

Les premiers algorithmes pour la coloration des graphes ont été développés dans les années soixante [**Welsh and Powell**, 1967; **Christofides**, 1971; **Brown**, 1972]. Dès lors, un nombre considérable de nouvelles techniques ont été développés et d’importants progrès ont été enregistrés. Tandis que des approches d’optimisation combinatoire à base d’énumération complète implicite (e.g. branch and bound) ont été testés pour la coloration exacte [**Brélaz**, 1979; **Mehrotra and Trick** 1996; **Sewell**, 1996; **Ramani**, 2006; **Lucet**, 2006; **Mendez-Diaz and Zabala**, 2006], très peu d’algorithmes exactes sont disponibles pour le problème [**Mendez-Diaz and Zabala**, 2006]. Les algorithmes exacts peuvent actuellement résoudre uniquement des graphes aléatoires de petite taille. Ce manuscrit s’articule autour de quatre chapitres, dont le premier est consacré au premier contact avec la théorie des graphes.

Le Chapitre deux est dédié à la Complexité algorithmique.

Le chapitre trois s’intéresse à la présentation, définition et types de la coloration.

Le Chapitre quatre est consacré à l’application des méthodes étudiées et des algorithmes de coloration avec implémentation en langage C (CodeBlocks).

Quelques notions élémentaires de la théorie des graphes

Les graphes permettent de modéliser toute situation dans laquelle il y a des interactions entre les objets. Les techniques utilisées en théorie des graphes permettent de répondre à beaucoup de problèmes algorithmiques posés, en effet, étudier les propriétés de ces problèmes revient à étudier les propriétés structurelles de leurs topologies représentées par des graphes. Dans ce chapitre, nous allons avoir un premier contact avec les graphes, sur les définitions et notations de base de la théorie des graphes, qui permettent à ce document d'être auto-contenu. On adoptera la terminologie de Berge [4]. D'autres définitions peuvent être retrouvées avec plus de détails dans les ouvrages de références pour la théorie des graphes, comme [5] [9]

1.1 Définitions et concepts de base

1.1.1 Graphe

un graphe G est constitué d'un ensemble V fini d'éléments $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$ appelés sommets, et d'une famille $E(G) = \{e_1, \dots, e_n\}$ de paires distinctes de V appelés arêtes.

La Figure 1.1 montre un exemple d'un graphe à 4 sommets et 4 arêtes.

1.1.1.1 Un graphe orienté

Définition 1.2.1 graphe orienté $G = (V, U)$ est défini par :

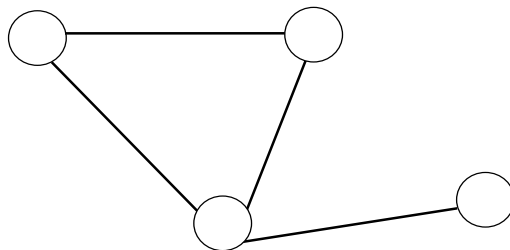


FIGURE 1.1 – Exemple d'un graphe à 4 sommets et 4 arêtes.

- Un ensemble V dont les éléments sont appelés des sommets ou des noeuds. Si le cardinal de V est égal à n ($|V| = n$) on dit que le graphe G est d'ordre n .
- Un ensemble U dont les éléments $u \in U$ sont des couples ordonnés de sommets appelés arcs de G . $u = (v_i, v_j)$; (v_G, v_i) , (v_i, v_G) est un arc de G , v_i est l'extrémité initiale de u et v_j est l'extrémité terminale de u .

$|U| = m$ est le nombre d'arêtes .

Le sommet v_i est un successeur de sommet v_G s'il existe un arc de la forme (v_G, v_i) et le sommet v_i est un prédécesseur du sommet v_G s'il existe un arc de la forme (v_j, v_i) .

La Figure 1.2 montre un exemple d'un graphe orienté à 5 sommets et 7 arêtes.

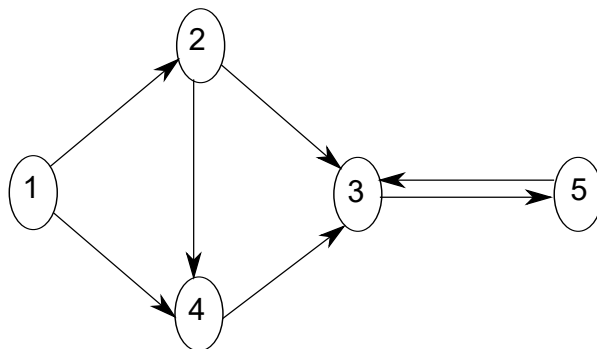


FIGURE 1.2 – Graphe orienté à 5 sommets et 7 arcs.

1.1.1.2 Un graphe non orienté

Définition 1.2.2 Un graphe non orienté $G = (V, E)$ est défini par :

- Un ensemble v dont les éléments sont appelés des sommets ou des noeuds .

- Un ensemble E dont les éléments $e \in E$ sont des couples non-ordonnés de sommets appelés des arêtes de G . $e = \{v_i, v_j\}$ est une arête de G , et v_i et v_j sont ces extrémités.

Remarque 1.2.1 Une arête peut être considérée comme étant deux arcs ; de sens opposés.

La Figure 1.3 montre un exemple d'un graphe non orienté à 4 sommets et 3 arêtes.

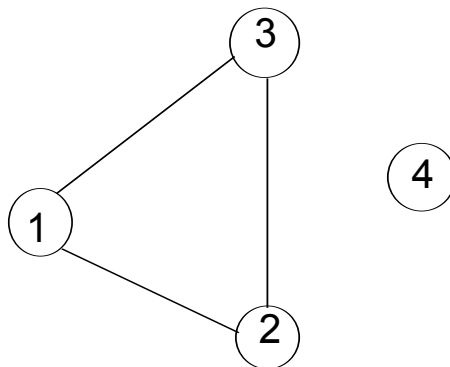


FIGURE 1.3 – Un Graphe non orienté à 4 sommets et 3 arêtes.

Dans notre étude nous allons considérer que les graphe non orienté.

Une **arête** e est une paire de sommets (u, v) qui relie deux sommets entre eux. Les sommets u et v sont les **extrémités** de l'arête.

Deux sommets d'un graphe sont **voisins** ou **adjacents** s'ils ont au moins une arête incidente en commun.

Dans un graphe G , une arête e reliant un sommet u à un sommet v est notée par uv et dans ce cas on dit :

- u et v sont adjacents
- u et v sont les extrémités de e ; e incidente à u et v
- Deux arêtes sont dites adjacentes si elles sont incidentes à un même sommet.

On appelle **ordre** d'un graphe le nombre de sommets de ce graphe ou le cardinal de son ensemble de sommets, noté $|V(G)|$, et **taille** d'un graphe est nombre d'arêtes ou d'arcs dans un graphe.

On appelle **degré** d'un sommet $u \in V(G)$, et on note $d_G(u)$, le nombre d'arêtes incidentes à u .
 $d_G(u) = |N(u)|$ et on note :

$$\delta(G) = \min_{u \in V} d_G(u) \text{ (degré minimum de } G)$$

$$\Delta(G) = \max_{u \in V} d_G(u) \text{ (degré maximum de } G)$$

Lorsque $d_G(u) = 0$, on dit que le sommet u est **isolé**.

Lorsque $d_G(u) = 1$, on dit que le sommet u est **pendant**.

Lorsque plusieurs arêtes relient deux sommets, on les appelle des arêtes **multiples**.

Une **boucle** est une arête dont les deux extrémités sont identiques.

La Figure 1.4 montre un graphe d'ordre 8 comportant 8 sommets ($V = \{1, 2, 3, 4, 5, 6, 7, 8\}$) et 9 arêtes ($E = \{(1,2), (1,3), (2,3), (2,4), (2,5), (2,6), (4,6), (6,7), (3,7)\}$)

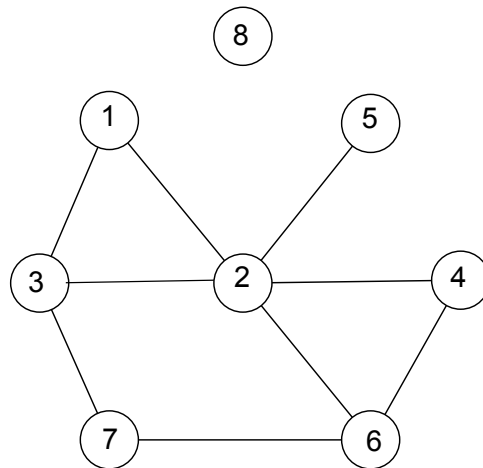


FIGURE 1.4 – Graphe à 8 sommets et 9 arêtes.

Dans la Figure 1.4 on a :

$$\delta(G) = 0$$

$$\Delta(G) = 5$$

1.1.2 Sous graphe

Soit $G = (V, E)$ un graphe, alors un graphe $G' = (V', E')$ est dit sous-graphe de G si :

$$- V' \subset V$$

$$- E' \subset E$$

$$- E' = \{(u, v) \mid (u, v) \in E \wedge u \in V' \wedge v \in V'\}$$

Un sous-graphe d'un graphe donné est donc obtenu en enlevant certains sommets, et toutes les arêtes incidentes à ces sommets. Voir le sous graphe (b) de graphe de la Figure 1.5 telle que :

$$V' = \{v_1, v_3, v_4, v_5\}$$

$$E' = \{e_3, e_4, e_5\}$$

1.1.3 Graphe partiel

Soit $G = (V, E)$ un graphe. Le graphe $G' = (V, E')$ est un graphe partiel de G , si E' est inclus dans E et $V' = V$.

Autrement dit, on obtient G' en enlevant une ou plusieurs arêtes au graphe G (sans toucher à ses sommets) voir le graphe partiel (c) de graphe de la Figure 1.5 Telle que :

$$V' = V$$

$$E' = \{e_5, e_4, e_3\}$$

1.1.4 Sous graphe partiel

Un graphe partiel d'un sous-graphe est un sous-graphe partiel de G .

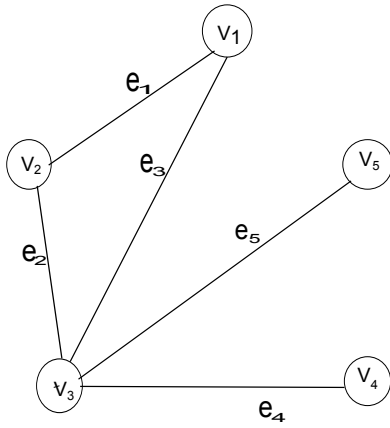
Cette fois-ci, on enlève des sommets (et leurs arêtes incidentes), puis des arêtes on obtient le sous-graphe partiel de graphe G voir le Sous graphe partiel (d) de graphe de la Figure 1.5 Telle que :

$$V' = \{v_1, v_2, v_3, v_4\}$$

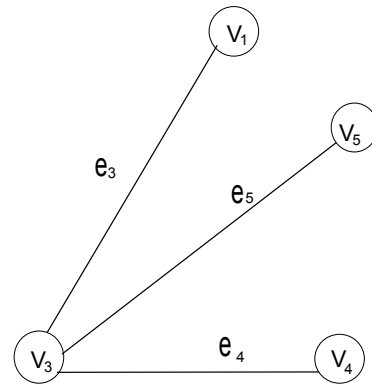
$$E' = \{e_1, e_4\}$$

1.1.5 Clique

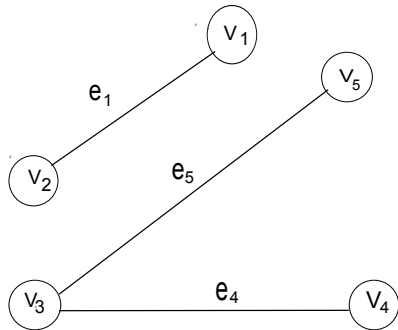
Une clique dans un graphe est un ensemble de sommets deux à deux adjacents.



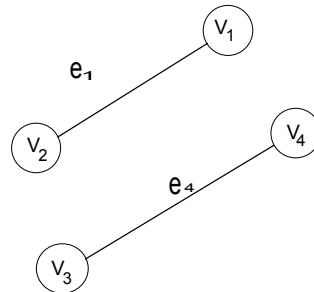
(a) Graphe G.



(b) Sous graphe de G.



(c) Graphe partiel de G.



(d) Sous graphe partiel de G.

FIGURE 1.5 – Les graphe iduite à partir de graphe G

La taille de la plus grande clique dans un graphe G se note $\omega(G)$. On a donc réussi, en enlevant des sommets, à faire en sorte que chaque sommet est adjacent à tous les autres sommets.

Le graphe (b) de la Figure 1.6 représente une clique de graphe G .

1.1.6 Stable

Un stable est un sous graphe induit G' tel qu'il n'existe pas d'arêtes entre les sommets de G' . L'ordre de plus grand stable d'un graphe G est noté $\alpha(G)$.

On enlève donc des sommets, et leurs arêtes adjacentes. On obtient un stable que si, en ayant enlevé un certain nombre de sommets à G , le sous-graphe résultant ne possède plus d'arête.

Le graphe (c) représente une stable de graphe G .

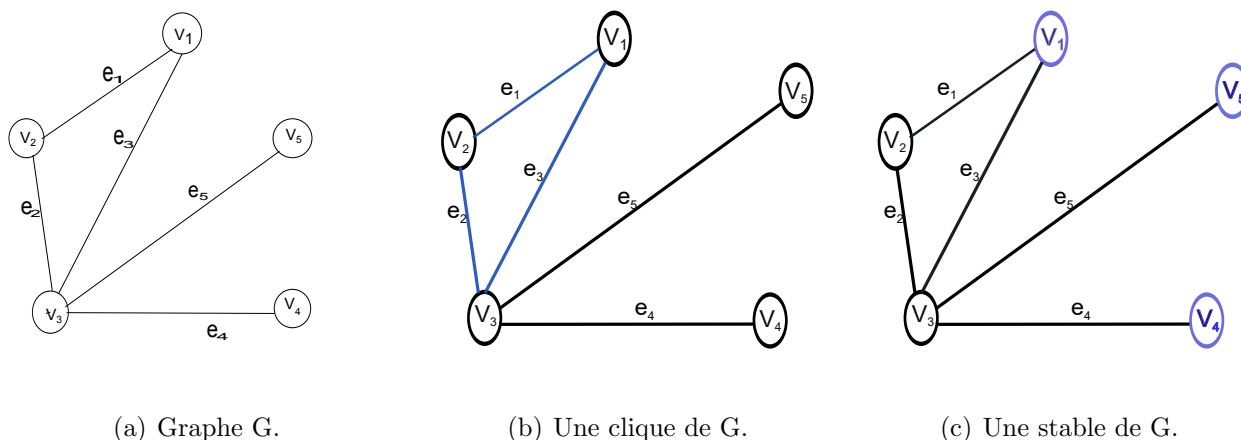


FIGURE 1.6 –

1.1.7 Couplage

Un couplage M d'un graphe $G = (V, E)$ est un ensemble d'arêtes deux à deux non adjacentes. Un sommet de graphe est dit saturé par le couplage M , si il est l'extrémité d'une arête de M , sinon il est dit insaturé si tout sommets de G est saturé alors le couplage est dit parfait.

- Un couplage est dit maximum s'il est de cardinalité maximum..
- Le couplage maximum est le couplage couvrant le plus grand nombre de sommets possibles, en laissant donc le moins de sommets isolés.

La Figure 1.7 représente un Couplage parfait.

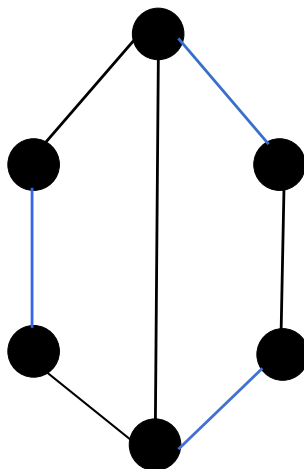


FIGURE 1.7 – Couplage parfait.

1.1.8 Transversal

Un transversal d'un graphe est un sous-ensemble de sommet V tel que toute arête du graphe est incidente à au-moins un sommet de V .

Le nombre transversal noté $\tau(G)$.

1.2 Chaîne et Cycle dans les graphe

Une chaîne entre deux sommets u et v d'un graphe G est une suite de sommets $u = u_1, u_2, \dots, u_k = v$; dont deux sommets consécutifs sont adjacents.

Une chaîne reliant deux sommets u et v de G sera dite une (u, v) -chaîne.

Une chaîne simple est une chaîne qui n'utilise pas deux fois la même arête.

Une chaîne élémentaire est une chaîne qui n'utilise pas deux fois le même sommet.

La longueur d'une chaîne est le nombre d'arêtes formant cette chaîne.

On appelle un cycle dans un graphe G , une chaîne simple dont les extrémités sont confondues. Une chaîne Hamiltonienne (resp. cycle Hamiltonien) est une chaîne (resp. cycle) utilisant tous les sommets une et une seules fois.

Un graphe $G = (V, E)$ possédant un cycle Hamiltonien est dit graphe Hamiltonien.

Une chaîne Eulérienne (resp. cycle Eulérien) est une chaîne (resp. cycle) utilisant toutes les arêtes une et une seule fois.

Un graphe $G = (V, E)$ qui possède un cycle Eulérien est dit graphe Eulérien.

1.3 Connexité dans les graphes

Définition 1.4.1 Un graphe est connexe si chaque sommet est accessible à partir de n'importe quel autre. Autrement dit, si pour tout couple de sommets distincts $(v_i, v_j) \in V^2$, il existe une chaîne entre v_i et v_j .

Le graphe de la Figure 1.8 n'est pas connexe car il n'existe pas de chaîne entre les sommets a et e . En revanche, le sous-graphe induit par les sommets $\{a, b, c, d\}$ est connexe.

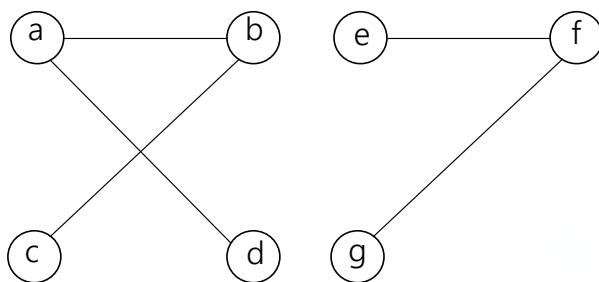


FIGURE 1.8 – Graphe non connexe (Sous graphe connexe)

Définition 1.4.2 Une composante connexe d'un graphe G est un sous-graphe G' de G qui est connexe et maximal (c'est-à-dire qu'aucun autre sous-graphe connexe de G ne contient G').

– Un graphe est dit connexe si et seulement si il admet une unique composante connexe.

Par exemple, le graphe de la Figure 1.8 est composé de 2 composantes connexes : la première est le sous-graphe induit par les sommets $\{a, b, c, d\}$, et la seconde est le sous-graphe induit par les sommets $\{e, f, g\}$.

1.3.1 Isthme

Dans un graphe connexe un isthme est une arête dont la suppression augmente le nombre de

composantes connexes ayant chacune au moins une arête. L'arête e du graphe de la Figure 1.9 est un isthme.

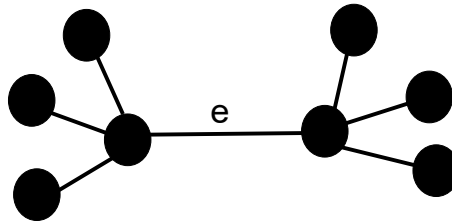


FIGURE 1.9 – Isthme.

1.4 Représentation des graphes

Un certain nombre de représentations existe pour décrire un graphe. On distingue principalement la représentation par matrice d'adjacence et d'incidence sommets-arêtes. linéaire peuvent également être utilisés pour coder les graphes.

Matrice d'adjacence

Considérons un graphe $G = (V, E)$ comportant n sommets. La matrice d'adjacence est égale à la matrice $M = (m_{ij})$ de dimension $n * n$ telle que :

$$M_{ij} = \begin{cases} 1, & \text{si il existe au moins une arête } v_i \text{ vers } v_j ; \\ 0, & \text{sinon.} \end{cases}$$

Matrice d'incidence

Soit un graphe $G = (V, E)$ avec $|V| = n$ et $|E| = m$, on appelle matrice d'incidence sommets arêtes, la matrice définit comme suit :

$$M_{ij} \begin{cases} +1, & \text{Si } v_i \text{ est l'extrémité initiale de } e_j ; \\ -1, & \text{Si } v_i \text{ est l'extrémité terminale de } e_j ; \\ 0, & \text{Si } v_i \text{ n'est pas une extrémité de } e_j . \end{cases}$$

Pour un graphe non orienté sans boucles, la matrice d'incidence (aux arêtes) est définie par :

$$M_{ij} = \begin{cases} 1, & \text{Si } v_i \text{ est une extrémité de } e_j; \\ 0, & \text{sinon.} \end{cases}$$

La Figure 1.10 est un exemple d'un graphe à 5 sommets et 6 arêtes :

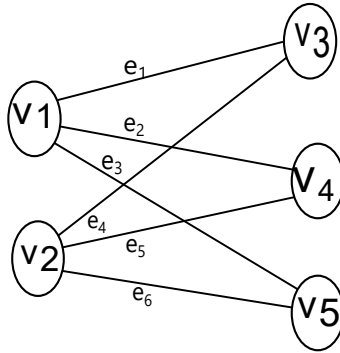


FIGURE 1.10 – Un Graphe G.

La matrice d'adjacence du graphe G représenté sur la Figure 1.10 est donné par :

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

La matrice d'incidence associée au graphe de la Figure 1.10 est :

$$B = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

1.5 Opération classiques sur les graphes

1.5.1 Somme Cartésienne de deux graphes

Étant donnés deux graphes $G = (V(G), E(G))$ et $H = (V(H), E(H))$, la somme Cartésienne de G avec H , notée $G \square H$ est le graphe défini sur l'ensemble de sommets $V(G) \times V(H)$ tel que deux sommets (u, u') et (v, v') sont adjacents si et seulement si l'une des propriétés suivantes est vérifiée :

- $u = v$ et $u'v' \in E(H)$
- ou
- $uv \in E(G)$ et $u' = v'$

La Figure 1.11 montre la somme cartésienne de $H = K_2 \square C_4$.

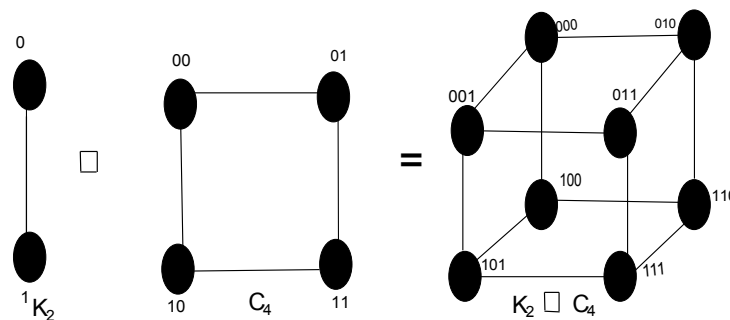


FIGURE 1.11 – Somme Cartésienne de $K_2 \square C_4$.

Il est noté que le nombre de sommets dans $G \square H$ est $|V(G)| \times |V(H)|$ et que le nombre d'arêtes de $G \square H$ est $|V(G)| \times |E(H)| + |V(H)| \times |E(G)|$.

1.5.2 Produit Cartésien de deux graphes

Étant donnés deux graphes $G = (V(G), E(G))$ et $H = (V(H), E(H))$, le produit Cartésien de G par H , noté $G \times H$ est le graphe défini sur l'ensemble de sommets $V(G) \times V(H)$ tel que deux sommets (u, u') et (v, v') sont adjacents si et seulement si $uv \in E(G)$ et $u'v' \in E(H)$.

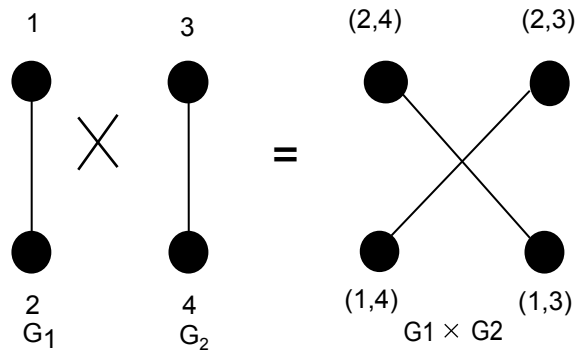


FIGURE 1.12 – Produit cartésien de deux graphes $G_1 \times G_2$.

La Figure 1.12 montre le produit cartésien de $K = G_1 \times G_2$.

1.5.3 Subdivisions de graphes

En remplaçant les arêtes d'un graphe $G = (X, E)$ par des chaînes disjointes intérieurement, on obtient une subdivision de G . Une subdivision d'une arête $u \rightarrow v$ s'obtient en ajoutant au graphe de départ de nouveaux points (x_1, \dots, x_n) et en remplaçant l'arête $u \rightarrow v$ par une chaîne $(u \rightarrow x_1 \rightarrow \dots \rightarrow x_n \rightarrow v)$ (la subdivision triviale obtenue pour $n = 0$ consiste à ne rien changer). Un graphe G' est une subdivision d'un graphe G si on l'obtient par subdivision distinctes de ses arêtes [6].

Exemple 1.6.1

La Figure 1.13 montre la Subdivision de graphes

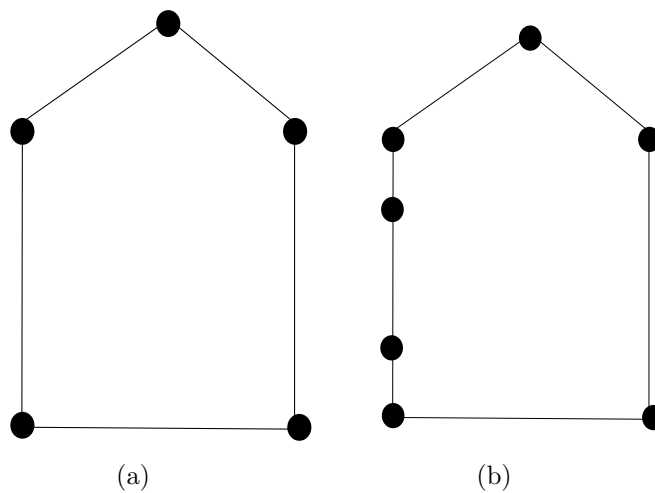


FIGURE 1.13 – Subdivision de graphes

1.5.4 Morphismes de graphes

Soient $G = (V(G), E(G))$ et $H = (V(H), E(H))$ deux graphes. Un homomorphisme de G dans H est une application f de $V(G)$ dans $V(H)$ telle que pour toute arête uv de G on a $f(u)f(v)$ est une arête de H c'est à dire $(\forall (u, v) \in V(G) \times V(G), uv \in E(G)) f(u)f(v) \in E(H)$.

La Figure 1.14 montre l'homomorphisme de G dans H .

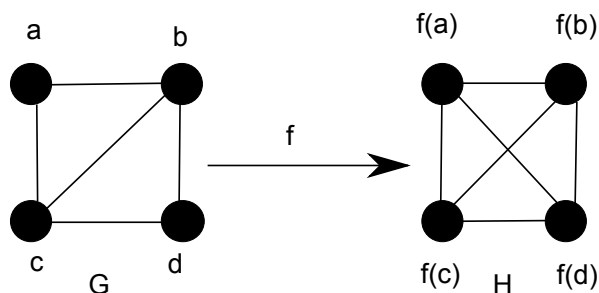


FIGURE 1.14 – Homomorphisme de G dans H .

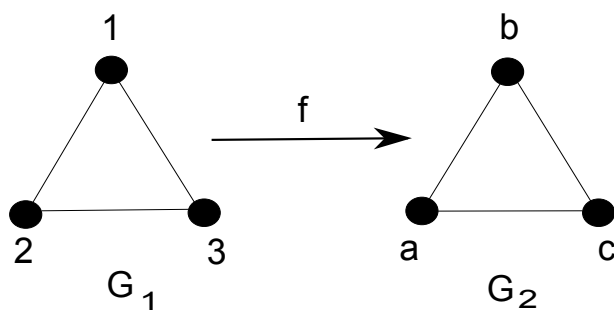
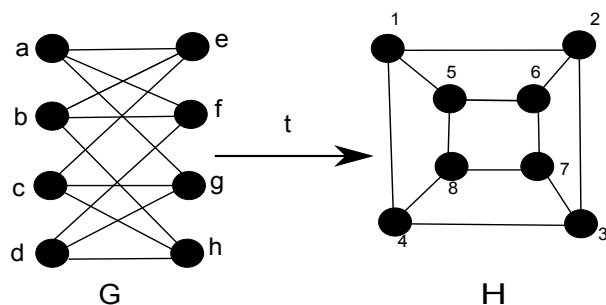
1.5.5 Isomorphismes

Deux graphes $G = (V(G), E(G))$ et $H = (V(H), E(H))$ sont dits isomorphes si et seulement s'il existe une application bijective $\varphi : V(G) \rightarrow V(H)$ qui vérifie la condition suivante :

$$uv \in E(G) \Leftrightarrow \varphi(u)\varphi(v) \in E(H).$$

Ceci signifie aussi que φ est un morphisme et φ^{-1} est un morphisme.

Un isomorphisme de G dans lui même est appelé Isomorphisme intérieur ou bien automorphisme. On voit facilement dans la figure 1.15(a), lorsque on prend l'application f de $V(G_1)$ dans $V(G_2)$, tels que $f(1) = b$, $f(2) = a$ et $f(3) = c$. f présente bien un homomorphisme, mais on ne peut pas trouver un homomorphisme g de $V(G_2)$ dans $V(G_1)$, donc f n'est pas un isomorphisme par contre dans la Figure 1.15(b), l'application t de $V(G)$ dans $V(H)$, tels que $t(a) = 1$, $t(b) = 6$, $t(c) = 8$, $t(d) = 3$, $t(e) = 5$, $t(f) = 2$, $t(g) = 4$ et $t(h) = 7$ montre que G et H sont isomorphes.

(a) Homomorphisme de G_1 dans G_2 

(b) isomorphisme entre G et H

FIGURE 1.15 – Homomorphisme de G_1 dans G_2 et isomorphisme de G dans H.

1.6 Distances et Intervalles dans les graphes

1.6.1 Distances dans les graphes

Étant donné deux sommets u et v d'un graphe $G = (V(G), E(G))$, on appelle distance entre u et v , la longueur d'une plus courte (u, v) -chaîne (en terme de nombre d'arêtes), une telle distance est noté $d_G(u, v)$ (ou $d_G(u, v)$ s'il n'y a pas de confusion). Une telle chaîne s'appelle géodésique.

- L'excentricité d'un sommet u noté $e_G(u)$ (ou $e(u)$ s'il n'y a pas de confusion) est le nombre suivant :

$$e_G(u) = \max d_G(u, v), v \in V(G).$$

- Le diamètre de G noté $D(G)$ (ou D s'il n'y a pas de confusion) est la plus grande excentricité :

$$D(G) = \max[e(u)], u \in V(G)$$

Un sommet $u \in V(G)$ est dit diamétral de $v \in V(G)$ si $d_G(u, v) = D(G)$. Si chaque sommet de G admet un unique sommet diamétral, on dira que G est diamétral.

- Le rayon de G noté $R(G)$ (ou R s'il n'y a pas de confusion) est la plus petite excentricité :

$$R(G) = \min[e(u)], u \in V(G)$$

- Le centre de G est l'ensemble des sommets de G dont l'excentricité est égale au rayon.
- La distance entre deux arêtes u_1v_1 et u_2v_2 dans un graphe $G = (V(G), E(G))$ est définie par :

$$d_G(u_1v_1, u_2v_2) = \min\{d_G(u_1, u_2), d_G(u_1, v_2), d_G(v_1, u_2), d_G(v_1, v_2)\}.$$

1.6.2 Intervalles

L'intervalle $I_G(u, v)$ (ou $I(u, v)$ s'il n'y a pas de confusion) c'est l'ensemble des sommets de G appartenant aux plus courtes (u, v) -chaînes. $I_G(u, v) = \{w \in V(G), \text{ avec } w \text{ est sur une plus courte } (u, v)\text{-chaîne}\}$. Trivialement, un sommet $w \in I(u, v)$ si et seulement si :

$$d_G(u, w) + d_G(w, v) = d_G(u, v).$$

1.7 Quelques graphes particuliers

1.7.1 Graphe simple

Un graphe G est dit simple s'il ne comporte pas de boucle, et si chaque deux sommets u et v il y a au plus une arête.

1.7.2 Graphe multiple

Si les boucles ou les arêtes multiples sont autorisés dans un graphe, on dira alors que le graphe est multiple.

1.7.3 Graphe K -régulier

Un graphe $G = (V, E)$ est dit k -régulier si :
 $v \in V$ on a $d_G(v) = k$.

1.7.4 Graphe complet

Un graphe $G = (V, E)$ est dit complet si tous les sommets sont deux à deux adjacents, le graphe complet à n sommets est noté K_n .

la Figure 1.16 montre le graphe complet K_5 .

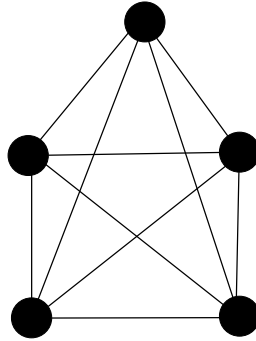


FIGURE 1.16 – Graphe complet K_5 .

1.7.5 P-graphe

Un graphe est un p -graphe s'il comporte au plus p arêtes entre deux sommets. Le plus souvent, on étudiera des 1-graphes.

1.7.6 Graphe biparti

Un graphe $G = (V, E)$ est dit biparti s'il existe une partition de l'ensemble des sommets de G en deux sous-ensembles V_1 et V_2 tel que toute arête de G a une extrémité dans V_1 et l'autre extrémité dans V_2 (deux sommets du même sous-ensemble ne soient jamais adjacents).

la Figure 1.17 montre un graphe biparti.

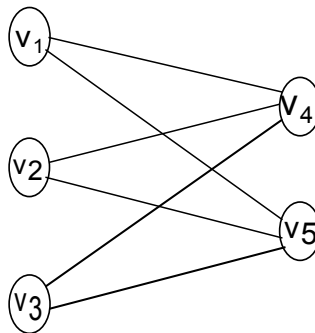


FIGURE 1.17 – Graphe biparti.

1.7.7 Graphe biparti complet

Un graphe biparti G est complet si tout sommet de V_1 est adjacent à tout sommet de V_2 . Si $|V_1| = p$ et $|V_2| = q$, on note $K_{p,q}$ ce graphe biparti complet. Un graphe $K_{p,q}$ admet $p + q$ sommets, $p \times q$ arêtes et le degré des sommets de V_1 (resp. V_2) est de q (resp. p).

1.7.8 Graphe biparti équilibré

Un graphe biparti G est dit équilibré si les deux sous-ensembles de la bipartition ont une même taille.

Les graphes ayant des cycles de longueur paire, i.e. les graphes bipartis complets, pour lesquels $p = q$, sont des exemples de graphes équilibrés.

1.7.9 Graphe planaire

Un graphe G est planaire si on peut le dessiner sur un plan sans qu'aucune arête ne se croise. Une application à ce type de graphe est le dessin de circuits électroniques.

La Figure 1.18 montre Graphe planaire et non planaire

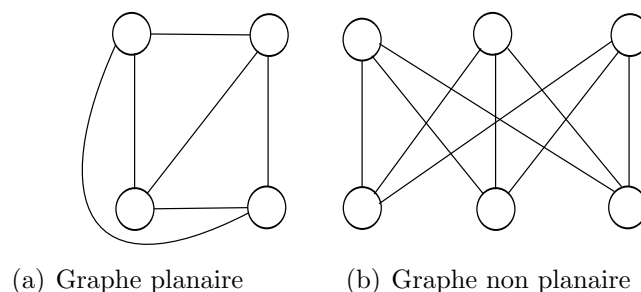


FIGURE 1.18 –

1.7.10 Arbres

Un arbre noté T est un graphe connexe sans cycle, on dira aussi qu'il existe une chaîne et une seule entre toutes paires de sommets.

Quelques propriétés des arbres

Théorème 1.8.1 [4]

Soit G un graphe d'ordre n . Les propriétés suivantes sont équivalentes.

1. G est sans cycle et connexe,
2. G est sans cycle et possède $n - 1$ arêtes,
3. G est connexe et possède $n - 1$ arêtes.
4. G est connexe et maximal pour cette propriété (lorsqu'on lui supprime une arête quelconque on va le déconnecter),
5. G est sans cycle et minimal pour cette propriété (lorsqu'on lui ajoute une arête on crée un cycle et un seul).
6. Tout couple de sommet (u, v) est relié par une chaîne et une seule.

Un graphe $G = (V(G), E(G))$ vérifiant au moins l'une des propriétés ci-dessus est un arbre T d'ordre n .

Théorème 1.8.2 [4]

Un arbre T admet au moins deux sommets pendants.

Arborescence

Le sommet v d'un graphe $G = (V, E)$ est une racine s'il existe dans G un chemin joignant v à x pour tous $x \in V$. Un graphe $G = (V, E)$ sur $n \geq 2$ sommets est une arborescence de racine v si :

- v est racine de $G = (V, E)$.
- G est un arbre .

Une arborescence est un arbre cependant la réciproque est fausse.

1.7.11 Étoile

Une étoile est un arbre avec exactement un sommet v qui n'est pas pendant, ce sommet est appelé jonction, et son degré est le nombre d'arêtes qui sont incidentes à v .

1.7.12 Chenille

une chenille C est un arbre avec au moins trois sommets, qui devient une chaîne P si tous les sommets pendants de cet arbre sont supprimés.

cette chaîne est appelée colonne de C , les sommets de P sont appelés sommets vertébraux (ou sommets épineux)

La Figure 1.19 montre une chenille

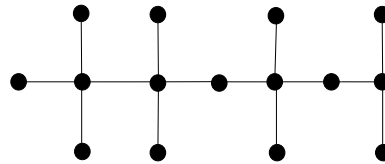


FIGURE 1.19 – Une chenille

1.7.13 Graphe valué (pondéré)

On dit qu'un graphe est valué, si on peut affecter des valeurs aux arêtes, ces valeurs peuvent représenter des coûts, probabilités ...etc.

1.7.14 Graphe de hamming

Les graphes de hamming est la somme cartésien de graphes complets sont une généralisation naturelle de l'hypercube.

Définition 1.8.1 Soient a_1, a_2, \dots, a_n des entiers positifs. Le graphe de Hamming H_{a_1, a_2, \dots, a_n} est le graphe dont l'ensemble des sommets est $\prod_{i=1}^n \{0, 1, \dots, a_i - 1\}$ et dont lequel deux sommets sont adjacents si et seulement si leur vecteur correspondant diffèrent exactement d'une seule composante.

On peut également le définir, comme étant la somme Cartésienne de n graphe complets :

$$H_{a_1, a_2, \dots, a_n} = K_{a_1} \square K_{a_2} \dots \square K_{a_n}$$

1.7.15 Graphe de Petersen

Le graphe de Petersen est, en théorie des graphes, un graphe particulier possédant 10 sommets et 15 arêtes.

Le graphe de Petersen n'est pas planaire. Tout graphe non planaire possède comme mineur soit le graphe complet $K_{\{5\}}$, soit le graphe biparti complet $K_{\{3,3\}}$, le graphe de Petersen possède les deux.

La Figure 1.20 montre le graphe de patersen

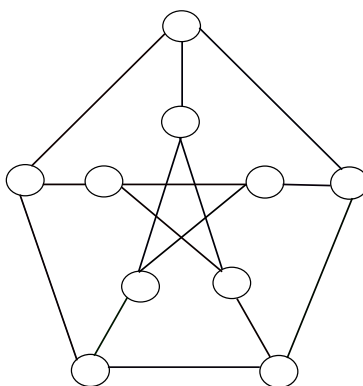


FIGURE 1.20 – Graphe de Petersen

1.7.16 Hypercube Q_n

Définition 1.8.1 L'hypercube de dimension n , noté Q_n est le graphe dont l'ensemble des sommets est formé des n -uplets binaires, et deux sommets sont adjacents si et seulement s'ils diffèrent exactement en une seule composante (coordonnée).

Notons que $Q_0 = K_1$, $Q_1 = K_2$ et d'une manière générale, Q_n peut être défini récursivement en utilisant la somme cartésienne par $Q_{n+1} = Q_n \square K_2$.

Il est clair que pour tout $n \geq 2$, Q_n est isomorphe à

$$\underbrace{k_2 \square k_2 \square \dots \square k_2}_{n \text{ fois}}$$

Donc

$$Q_{n+s} = Q_n \square Q_s$$

L'hypercube Q_n est construit récursivement à partir de deux copies disjointes Q'_{n-1} et Q''_{n-1} de Q_{n-1} et 2^{n-1} nouvelles arêtes comme suit :

Soit $V(Q'_{n-1}) = \{0U = 0u_2u_3\dots u_n\}$, ($u_i \in \{0,1\}$) et $V(Q''_{n-1}) = \{1V = 1v_2v_3\dots v_n\}$, ($v_i \in \{0,1\}$) tel que un sommet $0U$ est relié à un sommet $1V$ si et seulement si $u_i = v_i$ pour tout $i \in \{2, \dots, n\}$

La Figure 1.21 montre les premiers hypercubes.

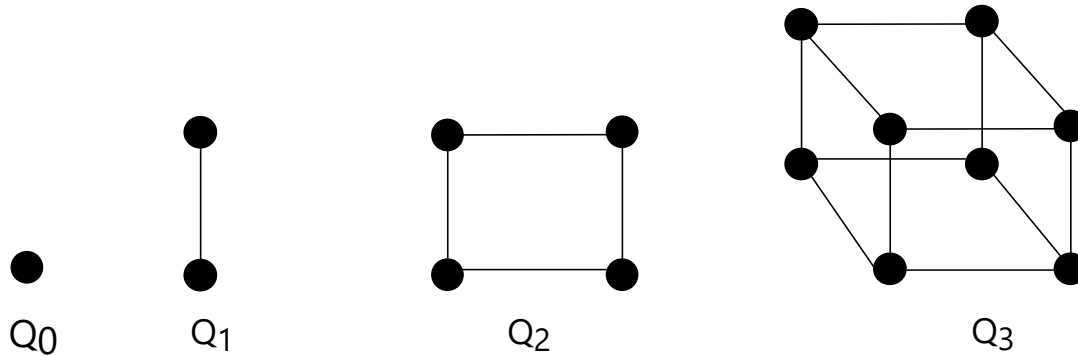


FIGURE 1.21 – Q_0, Q_1, Q_2, Q_3

Une direction i dans l'hypercube de dimension n ($i \leq n$) est l'ensemble des arêtes de Q_n , dont les extrémités ont des vecteurs associés qui diffèrent à la i -ème composante.

1.7.17 La Roue

En théorie des graphes, le graphe roue W_n est un graphe d'ordre $n \geq 4$ formé en ajoutant un sommet « centre » connecté à tous les sommets du graphe cycle C_{n-1} , Certains auteurs préfèrent W_{n-1} , faisant référence à la longueur du cycle.

La Figure 1.22 montre le graphe roue.

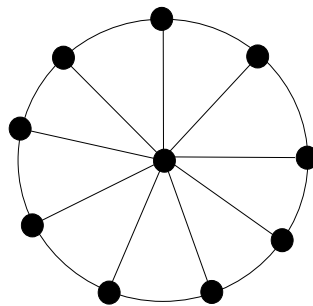


FIGURE 1.22 – Roue W_{10}

1.8 Conclusion

Dans ce chapitre nous avons donné un aperçu des notions de base de la théorie des graphes en s'appuyant sur quelques définitions et en l'illustrant avec quelques exemples.

Complexité algorithmique

2.1 Introduction

La théorie de la complexité est une branche des mathématiques et de l'informatique ayant pour cadre l'étude de la difficulté intrinsèque des problèmes algorithmiques, et qui vise à classer ces problèmes en fonction de cette difficulté. Ici, les mots « complexité » et « difficulté » ne se rapportent pas à la mise au point d'un algorithme de résolution, ou aux concepts avancés auxquels il peut faire appel (comme une structure de données élaborée), mais plutôt à la quantité de ressources à utiliser pour résoudre le problème.

Parmi les objectifs de la théorie de complexité est de classer les problèmes en fonction des ressources (temps de calcul, espace mémoire, etc...) nécessaires à leur résolution algorithmique. Ceci a montré qu'il existe des problèmes qui ont une solution calculable mais dont toute réalisation effective sur une machine est pratiquement inutilisable parce que le temps de calcul ou la place mémoire nécessaire sont trop importants.

2.2 Définitions

2.2.1 Algorithme

Un algorithme de résolution d'un problème P donné est une procédure décomposable en opérations élémentaire, transformant une chaîne de caractères représentant les données de n'importe quel exemple du problème P en une chaîne de caractères représentant les résultats de P.

2.2.2 Algorithmique

L'algorithmique est l'étude des algorithmes

2.3 Complexité d'un algorithme

Les problème de graphe font partie de la grande classe des problèmes d'optimisation combinatoire. On rencontre deux catégories de problèmes : les problèmes résolus rapidement par des algorithmes efficaces et les problèmes qui prennent un temps exponentiel sur certains cas complexes. ce sont respectivement des problèmes polynomiaux et exponentiels.

La notion de complexité d'un algorithme consiste à mettre en évidence les possibilités et les limites théoriques du processus calculatoire, elle est notée $O(O(n^2))$ pour une fonction qui augmente dans le carré de la taille des données). On rencontre aussi la notation Θ , $\Theta(n^2)$ qui donne une borne asymptotique par excès et par défaut (alors que O ne donne que la borne asymptotique par excès). [16]

Définition 2.3.1 (complexité algorithmique). Soit n la taille de la donnée en entrée, $f(n)$ une fonction définie sur l'entier positif. Un algorithme A est de complexité $O(f(n))$ dans le pire des cas s'il existe certaines constantes c et N_0 telles que :

$$\forall n > N_0; T_A(n) < c * f(n)$$

C'est-à-dire $T_A(n)$ elle est aussi vite que $f(n)$.

Définition 2.3.2 (Notations Ω , θ) [1]

Soient f et g deux fonctions $f; g : N \rightarrow R_+$.

- On note $f = \Omega(g)$ lorsque il existe un entier n_0 et une constante réelle c' tel que pour tout :

$$n > n_0, f(n) > c'g(n)$$
- On note $f = \theta(g)$ lorsque $f(n) = O(g)$ et $f(n) = \Omega(g(n))$, c'est-à-dire lorsque il existe un entier n_0 et deux constantes réelles c et c' tel que : $cg(n) \leq f(n) \leq c'g(n)$.

$\Omega(g)$ (respectivement $\theta(g)$) est l'ensemble des fonctions d'ordre inférieur (respectivement équivalentes) à g pour n assez grand.

Définition 2.3.3 (algorithme polynômial). Un algorithme A exécuté sur une entrée de taille n est dit polynômial s'il existe un entier i tel que A est de complexité $O(n^i)$.

Exemple 2.3.1

Supposons que le nombre d'opérations nécessaire à la terminaison d'un algorithme A sur une entrée de taille n est exactement $2n^2 - n - 7$, alors A est un algorithme polynômial. Au lieu de dire A est en $O(n^2)$, nous écrivons communément $T_A(n) = O(n^2)$.

Notons bien que la notion de complexité est utilisée pour spécifier une borne supérieure sur la croissance de la fonction. Elle n'est pas donnée de manière exacte. L'observation importante est que si la fonction du temps d'exécution est quadratique, alors si on double la taille d'entrée, le temps d'exécution va augmenter à quatre fois du temps actuel, et ceci ne dépend pas de la vitesse d'exécution de la machine.

Définition 2.3.4 (algorithme exponentiel). Un algorithme A exécuté sur une entrée de taille n est dit exponentiel s'il existe un réel r strictement supérieur à 1 et un polynôme $f(n)$ en n tel que A est de complexité de $O(r^{f(n)})$.

2.4 Calcul de complexité

2.4.1 Le temps d'exécution

Dans le calcul de temps d'exécution :

- chaque instruction basique consomme une unité de temps (affectation d'une variable, comparaison, addition, soustraction, multiplication, division,...).
- Chaque itération d'une boucle rajoute le nombre d'unités de temps consommées dans le corps de cette boucle.
- Chaque appel de fonction rajoute le nombre d'unités de temps consommées dans cette fonction.
- Pour avoir le nombre d'opérations effectuées par l'algorithme, on additionne le tout, on néglige le coût des déclarations, des affectations et du retour.

Par convention, on note n la taille des données.

On distingue les cas d'évaluation suivant :

- Dans le cas d'une séquence, le temps d'exécution total correspond à la somme des temps de traitements ;
- Dans le cas d'un branchement (test conditionnel) le temps total correspond au max des temps ;
- Dans le cas d'une boucle, cela correspond à la somme des coûts des passages successifs.

2.4.2 Règle de calcul simplificatif

D'après le calcul de temps d'exécution comme avant, on effectue les simplifications suivantes :

- on oublie les constantes multiplicatives(elles valent 1) ;
- On annule les constantes additives ;
- On ne retient que les termes dominants.

2.5 Différents types de complexité

En pratique, il y a évidemment une différence de temps de résolution entre les algorithmes polynômiaux et les algorithmes exponentiels. Pour donner un ordre d'idée sur les différentes complexités, le tableau ci-dessus présente les différents types de complexités et leur temps d'exécution. [15]

Les différences de temps nécessaires à la résolution de problèmes avec des complexités différentes peuvent être vues clairement dans le tableau 2.1, elles peuvent être phénoménales.

Complexité	Type de Complexité	Temps pour $n = 10$	Temps pour $n = 250$
$O(1)$	constante	$10ns$	$10ns$
$O(\log(n))$	logarithmique	$10ns$	$20ns$
$O((\sqrt{n}))$	racinaire	$32ns$	$158ns$
$O(n)$	linéaire	$100ns$	$2.5\mu s$
$O(n \log(n))$	linéarithmique	$100ns$	$6\mu s$
$O(n^2)$	quadratique (polynomial)	$1\mu s$	$625\mu s$
$O(n^3)$	cubique (polynomial)	$10\mu s$	$156ms$
$O(e^n)$	exponentielle	$10\mu s$	$10^{59}ans$
$O(n!)$	factorielle	$36ms$	$> 10^{100}ans$

TABLE 2.1-Évolution du temps de calcul en fonction de la complexité d'un algorithme et de la taille des données.

Plus précisément, les problèmes d'une complexité exponentielle ou factorielle sont impossibles à résoudre sur l'ensemble des données de taille raisonnable ($n > 250$).

2.6 Comment mesurer l'efficacité d'un algorithme

Un algorithme de résolution d'un problème P donné est une procédure, décomposable en opérations élémentaires, qui transforment les données en résultats. La première idée pour comparer et évaluer des algorithmes est de les programmer, puis de mesurer leurs durées d'exécution. Le temps de calcul dépend trop de la machine, du langage de programmation, du compilateur utilisé pour le langage et les données. En pratique on compte le " nombre d'opérations caractéristiques

” de l’algorithme à évaluer. Ce nombre ne dépend ni de la machine, ni du langage et peut s’évaluer sur du papier. L’évaluation dans certaines conditions s’appellent ” complexité de l’algorithme ”. Ce n’est pas la complexité de structure ou la difficulté de programmation.

2.7 La notion de problème

Les problèmes de graphe se rattachent à la grande classe des problèmes d’optimisation combinatoire. Tous ces problèmes se répartissent en deux catégories : ceux qui sont résolus optimalement par des algorithmes efficaces (rapides), et ceux dont la résolution peut prendre un temps exponentiel sur les grands cas.

2.7.1 Un problème en général

Définition 2.7.1 Un **problème** est une question générale possédant des paramètres dont la valeur n’est pas connue.[23]

Définition 2.7.2 Une **instance** d’un problème est obtenue en affectant une valeur à chacun de ses paramètres.

La **taille** d’une instance désigne généralement la quantité de cases mémoires nécessaires pour décrire les paramètres.[23]

2.7.2 Problèmes de décision et d’optimisation

Problème de décision

Un problème de décision P est un problème pour lequel l’ensemble des réponses S est $S = \{oui; non\}$.

L’ensemble des instances peut être partitionné en un ensemble D^+ d’instances positives, pour lesquelles la réponse est oui, et un ensemble D^- d’instances négatives, pour lesquelles la réponse est non.

Exemple 2.7.1

- Instance : un graphe orienté $G = (V; E)$, deux sommets $u; v \in V$.
- Question : Existe-t-il un chemin entre u et v dans G .

\Rightarrow C’est un problème de décision.

Problèmes d'optimisations

Un problème d'optimisation combinatoire (on dit aussi l'optimisation discrète) consiste à trouver dans un ensemble discret un parmi les meilleure sous-ensembles (ou solution) réalisables. Cet ensemble est fini mais compte un grand nombre d'éléments, d'où l'utilisation du terme "combinatoire". La notion de meilleure solution est définie au sens d'une fonction, qu'on appellera fonction objectif (ou parfois fonction de coût). Mathématiquement, cela peut se formuler de la manière suivante [12].

Soit S un ensemble fini et discret de solutions. Soit une fonction f telle que : $f : S \rightarrow R$. Un problème combinatoire consiste à trouver une solution s_0 minimisant (resp. maximisant) f , c'est à dire telle que $f(s_0) = \min_{s \in S} \{f(s)\}$ (resp : $f(s_0) = \max_{s \in S} \{f(s)\}$). Une autre manière d'écrire un problème d'optimisation est :

$$\min(\max)\{f(x) : x \in S\}$$

Exemple 2.7.2

Parmi les problèmes d'optimisation combinatoire classiques, on peut citer le problème du plus court chemin dans un graphe qui consiste à trouver, étant données un graphe et une valuation sur les arêtes du graphe, le chemin qui minimise la somme des valuations des arêtes reliant deux sommets du graphe. On peut aussi citer le problème du voyageur de commerce (abrégé en *TSP* depuis son nom anglais) est parmi les plus connus, et il consiste à trouver dans un graphe G un cycle élémentaire passant par tous les sommets et dont le coût est minimum. Enfin, le problème du stable maximum dans un graphe G , dans lequel on cherche un stable S de G tel que S soit de poids maximum.

Le problème du voyageur de commerce (ou **TSP** pour Traveling salesman problem) consiste, étant donné une liste de ville, et des distances connue entre toutes les paires de villes, détermine le plus court chemin qui relie toutes les villes, en ne passant qu'une seule fois par chaque ville. Une instance du **TSP** est donc un ensemble de n points (représentant les villes) définis chacun par un couple de coordonnées et la taille de cette instance est $2n + 1$ (il faut une case mémoire pour chaque coordonnée des n points et une autre pour stocker l'entier n).

2.8 Classes des problèmes

Nous allons maintenant nous intéresser à l'étude de la difficulté intrinsèque des problèmes de décision, ce que l'on appelle complexité des problèmes, et on va les classer selon la complexité des algorithmes. Un grand nombre d'entre eux sont des problèmes faciles car on connaît des algorithmes polynômiaux pour les résoudre.[23]

Un problème facile : Tout problème possédant une solution de complexité polynomiale est considéré facile

Un problème difficile : Une solution exponentielle, ou pire qu'exponentielle, est associée à un problème difficile.

Si le problème de reconnaissance associé à un problème POC donné est difficile alors le POC est dit difficile.

2.8.1 Les classes P (Polynomial time) et NP (Non deterministic Polynomial time)

Définition 2.8.1 La classe P est l'ensemble de tous les problèmes de décision pour lesquels il existe un algorithme polynomial. [23]

Exemple 2.8.1

Prenons par exemple le problème de la connexité dans un graphe

- Entrée : un graphe G à n sommets.
- Sortie : il s'agit de savoir si toutes les paires de sommets sont reliées par un chemin.

Pour le résoudre, on dispose de l'algorithme de parcours en profondeur qui va construire un arbre couvrant du graphe à partir d'un sommet. Si cet arbre contient tous les sommets du graphe, alors le graphe est connexe. Le temps nécessaire pour construire cet arbre est au plus n^2 , donc le problème est bien dans la classe P .

Les problèmes de la classe P sont dits faciles. Ce sont ceux que l'on sait résoudre efficacement.

Définition 2.8.2 Un algorithme non déterministe est muni (à l'inverse des déterministes) d'une instruction qui permette, chaque fois qu'elle est appliquée, de faire le bon choix.

Définition 2.8.3 Un problème Π appartient à la classe NP si Π peut être résolu par un algorithme polynomial non déterministe. NP signifie non déterministe polynomial. Supposons qu'on sait que la réponse à un problème de reconnaissance est vraie. Si on peut faire partager notre conviction à une autre personne en temps polynomial, alors le problème appartient à la classe NP , même si on ne sait trouver en temps polynomial une solution pour laquelle la réponse est vraie.

Un problème Π appartient à la classe NP est un problème qu'on peut résoudre par une exploration arborescente dont la profondeur, la longueur du plus long chemin de la racine à un sommet pendant, soit une fonction polynomiale de la taille des données. On conjecture que $P \neq NP$. La

classe NP contient des problèmes qui sont plus difficiles que ceux de la classe P si toutefois $P \neq N$.

Exemple 2.8.2

Considérons la version "décision" du problème du stable :

- Entrée : un graphe $G = (V; E)$ et un entier positif k .
- Sortie : existe-t-il un stable de taille au moins k ?

Ce problème est clairement dans NP : si l'on dispose d'un ensemble S de sommets, on peut vérifier en temps polynômial que $|S| \geq k$ et que S est stable (par exemple en examinant la liste d'adjacence de chaque sommet de S).

Remarque

1. Les problèmes de la classe NP sont ceux que l'on peut résoudre par énumération complète de toutes les solutions possibles (méthode "brutale") et en les testant à l'aide d'un algorithme polynômial.
2. On a clairement $P \subseteq NP$. En effet, si on peut résoudre un problème par un algorithme polynômial, alors on peut aussi vérifier en temps polynômial que la solution fournie est bien une solution du même problème.
3. La question de savoir si $P = NP$ est un problème ouvert, le plus important, de la théorie de la complexité. Cela revient à savoir si le fait de chercher une solution est aussi simple que de vérifier une solution. De nombreuses personnes pensent que $P \neq NP$.

2.8.2 La classe NP-complet

Un problème NP est NP-complet si tout problème NP s'y réduit en temps polynomial. Autrement dit, les problèmes NP les plus difficiles sont NP-complets.

propriété

- Si un seul problème NP-complet peut être résolu en un temps polynomial, alors tous les problèmes de NP peuvent être résolus en un temps polynomial.
Un problème NP-complet est un problème dans NP au moins aussi difficile que tout autre problème de NP.
- Il existe une réduction polynomiale qui permet de décrire n'importe quelle instance de n'importe quel problème de NP comme une instance de NP-complet.
- Si on sait résoudre toutes les instances d'un problème NP-complet on sait résoudre toutes les instances de tous les problèmes NP.

Par exemple

Ce problème consistant à élever un nombre au carré se réduit au problème plus général de multi-

plication de deux nombres (ici, aucune transformation n'est nécessaire). Une réduction est polynomiale lorsque le processus de transformation peut se faire en temps polynomial.

2.9 Conclusion

Dans ce chapitre, on s'est intéressé aux éléments de la théorie de la complexité en donnant la définitions de la complexité d'un algorithme.

Coloration dans les graphes

3.1 Introduction

La coloration des graphes est un problème de la théorie des graphes est lié à de nombreuses applications répandues dans des domaines variés, tels que :

- L'affectation de fréquences dans les réseaux cellulaires.
- Les emplois du temps.
- La gestion de chaînes logistiques.
- La gestion du trafic aérien.
- Stockage de produits chimiques qui peuvent exploser s'ils sont en contacts (combien de wagons ou d'aires de stockage nécessaires?)
- Un cartographe qui doit colorier deux pays limitrophes avec deux couleurs différentes .

3.2 Définitions

3.2.1 Coloration des sommets

Colorier un graphe consiste à affecter une couleur à chacun de ses sommets de façon à ce que deux sommets adjacents ne portent pas la même couleur. On peut aussi définir une coloration d'un graphe de la manière suivante :

La coloration des sommets d'un graphe $G = (V, E)$ est une fonction v dans $c(v)$ associant à tout sommet $v \in V$ une couleur $c(v)$, en s'assurant que $c(v) \neq c(u)$ pour toute arête $[u; v] \in E$.

Le plus petit nombre de couleurs nécessaire pour colorer les sommets d'un graphe G est appelé "le nombre chromatique" de G et est noté $\chi(G)$.

- Une coloration utilisant k couleurs est dite k -coloration.
- Une coloration de sommets d'un graphe avec k couleurs est aussi une partition de l'ensemble des sommets en k stables.
- Il nous sera parfois commode de numéroter les couleurs d'une k -coloration de 1 à k , ou encore plus simplement de considérer que l'ensemble des couleurs est l'ensemble $\{1, \dots, k\}$.

3.2.2 Coloration des arêtes

Pour colorier les arêtes d'un graphe, on peut se ramener au problème de la coloration des sommets. Il suffit pour cela de travailler non pas sur le graphe lui-même, mais sur le graphe adjoint, noté G' , et que l'on définit ainsi :

- à chaque arête de $G = (V, E)$ correspond un sommet de $G' = (V', E')$
- deux sommets de G' sont reliés par une arête si les deux arêtes correspondantes de G sont adjacentes.

La Figure 3.1 montre la Coloration des arêtes

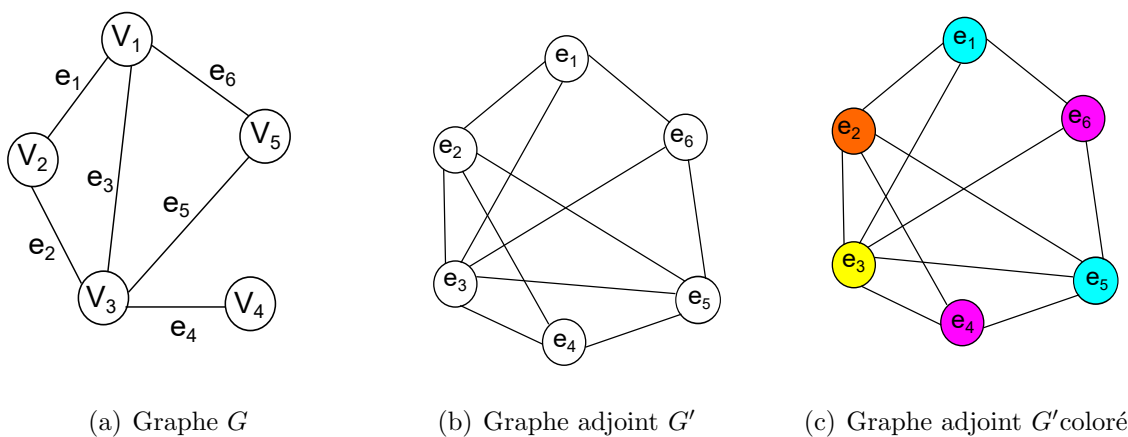


FIGURE 3.1 – La Coloration des arêtes

3.2.3 Nombre chromatique

Le nombre chromatique d'un graphe, noté $\chi(G)$, est le plus petit entier k pour lequel il existe une partition de V en k sous-ensembles stables c'est-à-dire le plus petit nombre de couleurs permettant de colorier tous les sommets du graphe.

3.3 Types de coloration

Dans la recherche il existe plusieurs types de coloration, considérons les types suivants :

3.3.1 Coloration simple

Soit $G = (V, E)$ un graphe donné, une coloration simple est une coloration des sommets, tel qu'il existe au moins deux couleurs i et j telles que le nombre de sommets colorier par i est différents du nombre de sommets colorier par la couleur j .

La Figure 3.2 présente une coloration simple d'un graphe à 5 sommets.

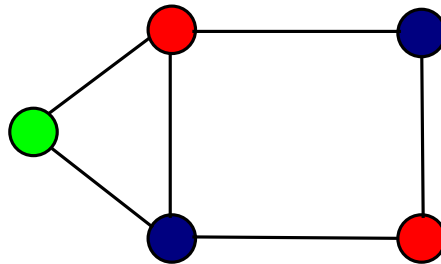


FIGURE 3.2 – Coloration simple.

3.3.2 Coloration K-équitable

Soit $G = (V, E)$ un graphe donné. une coloration K-équitable est une coloration des sommets du graphe G où le nombre de sommets coloriés par la couleur i est égale au nombre de sommets coloriés par la couleur j $\forall (i, j) \in 1, \dots, k$.

La Figure 3.3 présente une coloration K -équitable d'un graphe à 6 sommets.

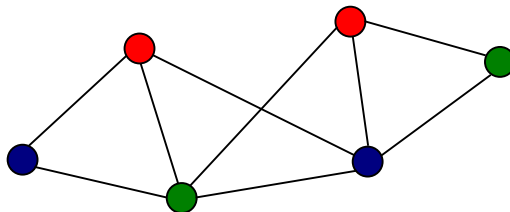


FIGURE 3.3 – Coloration 3-équitable.

Dans certains graphes le nombre chromatique est difficile à déterminer, donc la majorité des chercheurs ont étudiés l'encadrement du nombre chromatique.

3.3.2.1 Encadrement du nombre chromatique

1. Majoration : [21]

- Le nombre chromatique d'un graphe G est inférieur ou égal à $r + 1$.

$$\chi(G) \leq (r + 1) \dots (1).$$

Preuve : Soit G un graphe et r le degré maximum de ses sommets. Donnons-nous une palette de $(r + 1)$ couleurs. Pour chaque sommet du graphe on peut tenir le raisonnement suivant : ce sommet est adjacent à r sommets au plus, et le nombre de couleurs déjà utilisées pour colorer ces sommets est donc inférieur ou égal à r .

Il reste donc au moins une couleur non utilisée dans la notre coloration $(r + 1)$, avec laquelle nous pouvons colorer notre sommet.

$$\chi(G) \leq n + 1 - \alpha(G) \dots (2)$$

La Figure 3.4 présente un exemple qui vérifie la relation (1)

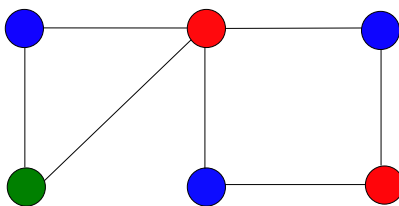


FIGURE 3.4 – Exemple 1.

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

La Figure 3.5 présente un exemple qui vérifie la relation (2)

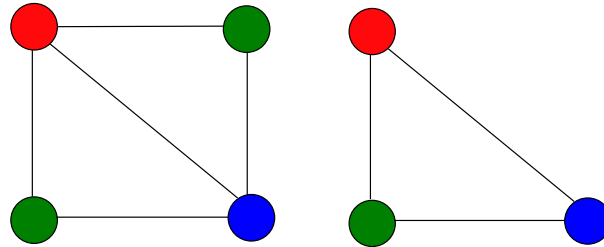


FIGURE 3.5 – Exemple 2.

2. Minoration : [21]

- Le nombre chromatique d'un graphe est supérieur ou égal à celui de chacun de ses sous-graphes.

Preuve : Ce résultat découle de la définition même du nombre chromatique.

- Le nombre chromatique du graphe sera supérieur ou égal à l'ordre de sa plus grande clique, que l'on note $\omega(G)$. Autrement dit, $\chi(G) \geq \omega(G)$

Preuve : Puisque, par définition, dans une clique d'ordre m , tous les sommets sont adjacents entre eux, il faudra m couleurs. Donc, il faudra au moins $\omega(G)$ couleurs pour colorer le graphe G .

Le nombre chromatique de quelques types de graphes : [3]

- Le nombre chromatique d'un arbre d'ordre $n \geq 2$ est 2.
- Le nombre chromatique d'un cycle d'ordre pair est 2 et d'ordre impair est 3.
- Le nombre chromatique d'une roue ayant un nombre pair de sommets est 4.
- Le nombre chromatique d'une étoile d'ordre $n \geq 2$ est 2

- Le nombre chromatique d'un graphe sans cycle ayant au moins une arête est 2. En effet, le nombre chromatique d'un graphe est le maximum des nombres chromatiques de ses composantes connexes, qui sont ici des arbres.
- Tout graphe complet d'ordre $n \geq 2$ est n -coloriable.

Remarque

Si un graphe à h composantes connexes, son nombre chromatique est le plus grand des nombres chromatiques de ces h composantes. L'étude du nombre chromatique peut donc se restreindre à celle du nombre chromatique des graphes connexes.

3.3.2.2 Théorème de Brooks(1941) [6]

Pour tout graphe connexe G qui n'est pas complet et qui n'est pas un cycle impair on a :

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

Proposition 3.3.1 [4]

Soit $G = (V, E)$ un graphe simple et $\alpha(G)$ le nombre de stabilité de G alors $\chi(G) \geq \alpha(G)$.
En résumé pour tout graphe G d'ordre n on a :

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

Proposition 3.3.2 [6]

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

$$\lfloor \frac{n}{\alpha(G)} \rfloor \leq \chi(G) \leq \Delta(G) + 1$$

Proposition 3.3.3 [6]

Soit $G = (V, E)$ un graphe simple d'ordre n Alors :

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

Pour obtenir une majoration du nombre chromatique des graphes planaires, plusieurs théorèmes ont été énoncés quant au nombre chromatique des graphes planaires.

3.3.2.3 Théorèmes de coloration des graphes planaires.

Voici un premier théorème permettant de colorer tout graphe planaire. **Aigner et Ziegler** [2] nous donnent une preuve simple de ce théorème important.

Théorème 3.3.1 : (Théorème des cinq couleurs) [2] Tout graphe planaire peut être coloré avec cinq couleurs.

Démonstration : Remarquons tout d'abord qu'ajouter des arêtes à un graphe ne peut qu'augmenter son nombre chromatique. Par conséquent, nous pouvons supposer que $G = (V; E)$ est connexe et que toutes les faces intérieures du graphe sont bornées par un triangle, c'est-à-dire 3 arêtes forment la frontière d'une face.

La Figure 3.6 illustre cette construction.

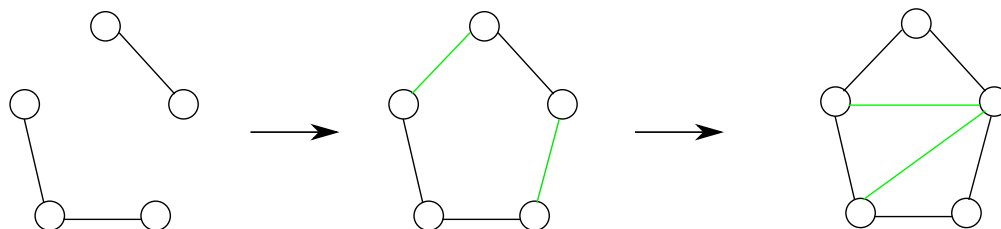


FIGURE 3.6 – Graphe auquel des arêtes ont été ajoutées pour obtenir un graphe connexe dont toutes les faces intérieures sont bornées par un triangle.

Il existe un théorème beaucoup plus fort que celui des cinq couleurs, démontré par **Appel et Haken** [14] [13] en 1976. Jusqu'ici, aucune preuve de ce théorème qui ne fait pas appel à l'ordinateur n'a été découverte.

En effet, 1478 cas critiques sont étudiés via l'utilisation d'un ordinateur. Ceci explique pourquoi aucune preuve n'est présentée dans cet partie.

Théorème 3.3.2 : (Théorème des quatre couleurs) [8] Tout graphe planaire peut être coloré avec quatre couleurs

Nous savons donc que tout graphe planaire admet une 4-coloration. Cependant, savoir si un graphe planaire est 3-coloriable reste un problème difficile (NP-complet). Étrangement, il est facile de connaître si le graphe peut être coloré avec 2 couleurs, autrement dit si le graphe est biparti.

Remarque : Le Théorème des quatre couleurs n'est pas efficace, il marche pas pour tout les carte géographique.

3.3.3 Coloration avec Satisfaction de Contraintes

Définition 3.3.1 (CSP) [22]

Les problèmes de satisfaction de contraintes (CSP) sont définis par un quadruplet (X, D, C, R) avec :

- $X = \{x_1, x_2, \dots, x_n\}$ est un ensemble fini de variables ;
- $D = \{d_1, d_2, \dots, d_n\}$ est un ensemble fini de valeur. Quel que soit $i \in \{1, \dots, n\}$, d_i est l'ensemble des valeurs pouvant être attribués à la variable x_i ;
- $C = \{c_1, c_2, \dots, c_m\}$ un ensemble fini de contraintes. Chaque contrainte c_i est définie par un uplet $(x_{i1}, x_{i2}, \dots, x_{ik})$ de k variables de X liées par c_i . On nomme $scope(c_i)$ l'ensemble $\{x_{i1}, x_{i2}, \dots, x_{ik}\}$;
- $R = \{r_1, r_2, \dots, r_m\}$ est un ensemble fini de relations associées aux m contraintes de C . Chaque relation r_i représente un ensemble de n -uplets de valeurs autorisées par la contrainte c_i .

Cette définition met en évidence l'ensemble des relations R qui est souvent implicite dans les définitions de la littérature.

Définition 3.3.2 (arité) L'arité d'une contrainte c est le nombre de variables sur lesquelles elle est définie. On la note $|c|$. Une contrainte est dite unaire (respectivement binaire) si elle est d'arité un (respectivement deux). Dans les autres cas, on la qualifie de n -aire, où n est l'arité de la contrainte.

Définition 3.3.3 (CSP binaire) Un CSP dont toutes les contraintes sont unaires ou binaires, est appelé un CSP binaire.

Définition 3.3.4 (CSP n-aire) Un CSP est n -aire s'il comporte des contraintes d'arité supérieure à deux.

Exemple d'un CSP (coloriage d'une carte)

L'objectif consiste à colorier une carte, de sorte que deux régions ayant des frontières en commun soient coloriées avec des différentes couleurs.

Ce problème peut être modélisé par le CSP suivant :

- $X = \{x_1, x_2, \dots, x_6\}$ est l'ensemble des régions à colorer ;
- $D = \{d_1, d_2, \dots, d_6\}$ est l'ensemble des domaines de chaque variable. pour tout i , on a :
 d_i est l'ensemble des couleurs pouvant être attribuées à la variable x_i et
 $d_i = \{\text{rose}, \text{jaune}, \text{bleu}\}$;
- $C = \{c_1, c_2, \dots, c_{10}\}$ est l'ensemble fini des contraintes. Chaque c_i est définie par un couple (x_{i1}, x_{i2}) de variables de X représentant des régions voisines sur la carte.
 $c_1 = (x_1, x_2), c_2 = (x_1, x_3), \dots, c_{10} = (x_5, x_6)$
- $R = \{r_1, r_2, \dots, r_{10}\}$ est l'ensemble des valeurs autorisées respectivement par $\{c_1, c_2, \dots, c_{10}\}$.

Chaque relation r_i représente l'ensemble de couples de couleurs différentes. Donc quel que soit i , $r_i = \{(\text{rose}, \text{jaune}), (\text{rose}, \text{bleu}), (\text{jaune}, \text{rose}), (\text{jaune}, \text{bleu}), (\text{bleu}, \text{rose}), (\text{bleu}, \text{jaune})\}$;

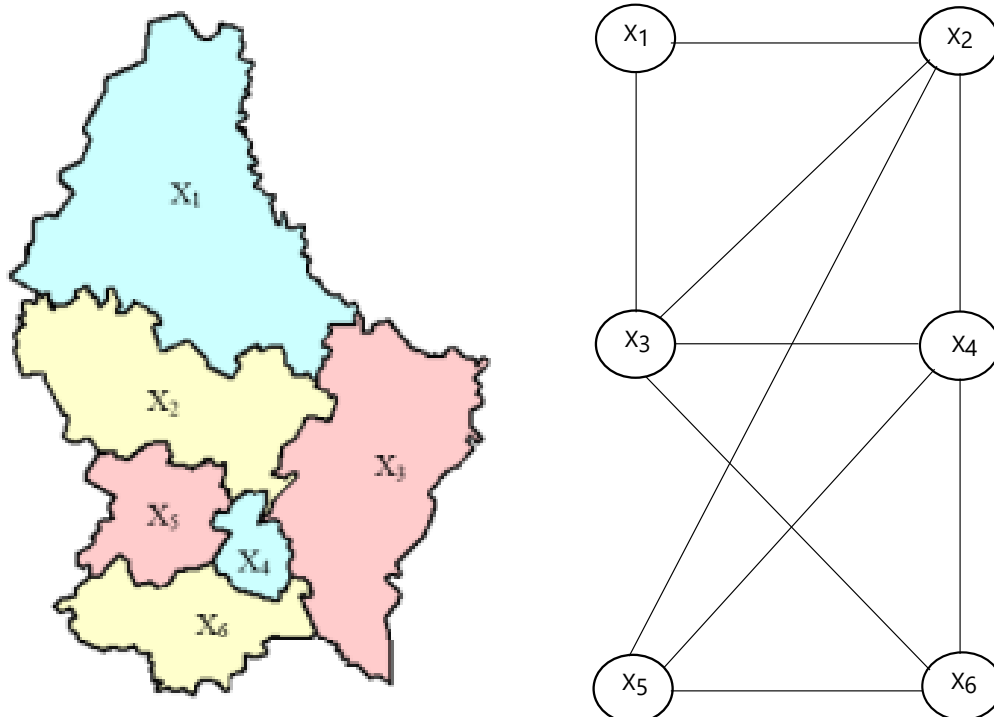


FIGURE 3.7 – Problème de coloriage de carte et son graphe des contraintes associé.

La Figure 3.7 illustre cet exemple et sa représentation graphique. Les domaines qui sont les couleurs possibles de chaque variable n'apparaissent pas dans cette représentation graphique (il existe une représentation en microstructure qui permet de voir les valeurs des domaines des variables en explicitant les n -uplets autorisés par chacune des contraintes). Les contraintes sont représentées par les arêtes du graphe. Elles expriment la différence des valeurs affectées aux variables.

Cette représentation ne fait pas apparaître le détail des domaines et des valeurs respectant chaque contrainte. Mais elle donne une idée générale de la structure du problème.

3.3.4 Coloration d'arêtes ℓ -distance

La coloration d'arêtes ℓ -distance a été définie en 1969 par Kramer et al [18], $\ell \geq 0$ est une généralisation de la coloration d'arêtes classique qui consiste à attribuer une couleur de 1 à k à chaque arête, telle que chaque deux arêtes qui sont à distance au plus ℓ ne partagent pas la même couleur.

Le nombre minimum de couleurs utilisées pour colorier un graphe avec une coloration d'arêtes ℓ -distance est appelé l'indice ℓ -chromatique et noté $\chi_\ell(G)$, ce résultat a été donné par Kramer.

Théorème 3.3.3 [17] Soit $G = (V, E)$ un graphe non orienté, fini connexe et sans boucle et ℓ un entier positif, $\ell \geq 2$ alors

$$\chi_\ell(G) = \ell + 1$$

si et seulement si l'une des conditions suivantes est remplie :

1. $|V| = \ell + 1$
2. E se réduit à un seul chemin de longueur $L \geq \ell$,
3. E se réduit à un seul cycle, dont la longueur est un multiple de $\ell + 1$.

3.3.4.1 Coloration d'arêtes ℓ -distance d'une chaîne

Théorème 3.3.4 [10] Soit P_n une chaîne d'ordre $n > 0$. L'indice ℓ -chromatique de P_n est donné par :

$$\chi'_\ell(P_n) = \begin{cases} \ell+2, & \text{si } n \geq \ell + 3; \text{(a)} \\ n-1, & \text{si } n \leq \ell + 3. \text{(b)} \end{cases}$$

Démonstration

cas (a) Dans un premier temps, nous démontrons par construction la borne supérieure de l'indice ℓ -chromatique. Soit Q l'ensemble de couleurs défini par $Q = \{1, 2, 3, \dots, \ell + 2\}$. Nous colorons d'une manière cyclique les arêtes de P_n en utilisant les couleurs de Q . Pour une coloration d'arêtes ℓ -distance d'un graphe G , la distance entre deux arêtes distinctes ayant la même couleur est égale au moins à $(\ell + 1)$. La coloration obtenue est une coloration d'arêtes ℓ -distance. Ainsi, $\chi'_\ell(P_n) \leq \ell + 2$. Dans un second temps, pour prouver la borne inférieure, soient e et e' deux arêtes de P_n telles que $dist(e, e') = \ell + 1$. Si nous supposons que ces deux arêtes ont la même couleur, toutes les arêtes intermédiaires qui relient e et e' auront des couleurs différentes. Il y a $\ell + 1$ arêtes entre e et e' . Ainsi, le nombre de couleurs est au moins égal à $(\ell + 1) + 1$ couleurs. Par conséquent, $\chi'_\ell(P_n) = \ell + 2$.

cas (b) Si $n \leq \ell + 3$, toutes les arêtes de P_n sont à une distance au plus ℓ les unes des autres. Par conséquent, toutes ces arêtes auront des couleurs différentes. D'où, $\chi'_\ell(P_n) = |E(P_n)| = n - 1$

La Figure 3.8 montre une coloration d'arêtes 1-distance de P_5 .

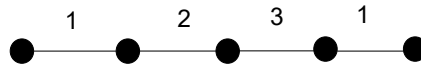


FIGURE 3.8 – Coloration d'arêtes 1-distance de P_5 .

3.3.4.2 Coloration d'arêtes ℓ -distance d'un hypercube [10]

la coloration d'arêtes ℓ -distance d'un hypercube consiste à Construction du sous-graphe $S_{\ell,d}$

Définition 3.3.5 Soient $\ell \geq 0, d \geq \ell + 1$ et Q_d un hypercube à d -dimensions. Le sous-graphe $S_{\ell,d} \subset Q_d$ est défini comme suit :

- $S_{\ell,d} \subset G \cup E'$ avec $G = Q_{\ell+1}$ et $E' \subset E(Q_d)$ sont des arêtes adjacentes à G

- pour tout $v \in V(Q_\ell)$ (avec $Q_\ell \subset G$ et v est aussi une extrémité de E'), le degré de v dans $S_{\ell,d}$ est égal à d et $|E'| = (d - \ell - 1)2^\ell$

La Figure 3.9 montre un exemple de construction de quelques sous-graphes $S_{\ell,d}$.

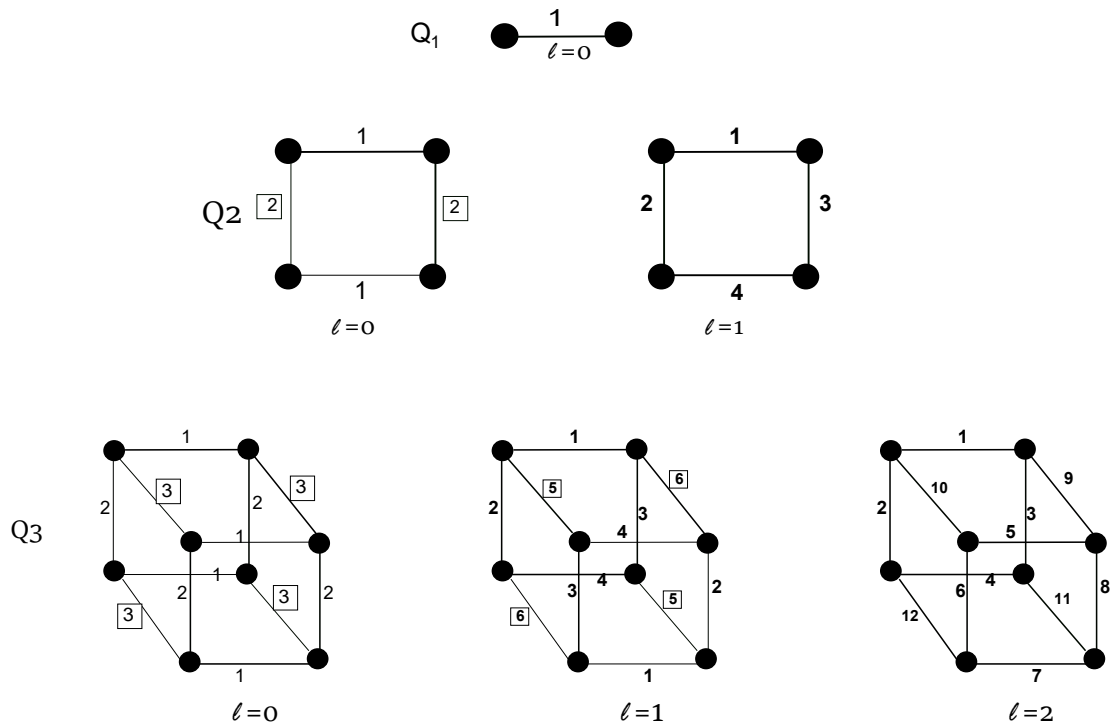


FIGURE 3.9 – Exemples de la coloration d’arêtes ℓ -distance de $Q_{i,1 \leq i \leq 4}$. Le sousgraphe $S_{\ell,d}$ est donne par les arêtes en gras. Les couleurs encadrées sont les couleurs des arêtes connecteurs.

3.3.5 Coloration par sommes sur les arêtes

Dans la coloration par sommes, il s’agit de mettre sur chaque arête d’un graphe G un poids de façon à ce que pour chaque sommet, la somme des poids des arêtes incidentes à ce sommet soit différente de celle de ses voisins. Autrement dit, la somme des poids des arêtes incidentes à un sommet lui donne une couleur et le but est d’obtenir une coloration propre des sommets (La coloration propre des arêtes qui permette de distinguer sommets adjacents via une fonction de codage $c, f(e)$ est le poide associe au arêtes telle que :

$$c(v) = \sum_{v \in e} f(e)$$

)

La Figure 3.10 donne un exemple de coloration par sommes pour le graphe de Petersen. Telle que sur le dessin, la somme des arêtes incidentes à un sommet est inscrit sur le sommet.

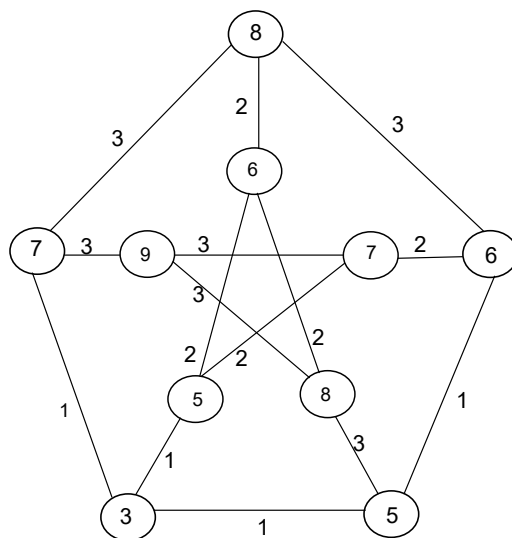


FIGURE 3.10 – Exemple de coloration par sommes sur les arêtes.

Ce type de coloration à été défini en 2004 par Karonski, luczak et Thomason qui ont émis la conjecture suivante :

Conjecture 1 (1-2-3 Conjecture [20]).

Pour tout graphe $G = (V, E)$ qui ne possédant pas d'arête isolée, il existe $K \in \{1, 2, 3\}$ pour lequel toutes les arêtes de G sont colorier par K et que le sommets adjacentes ne peuvent pas avoir le même poids.

Le graphe donné de la Figure 3.10 nécessite trois couleurs. Cela montre que le graphe de Petersen vérifie la Conjecture 1.

3.3.6 Coloration par multi-ensembles

Dans ce type de coloration, on attribue une couleur à chaque arête parmi un ensemble $\{1, 2, \dots, k\}$ de k couleurs. Pour chaque sommet v , on définit le multi-ensemble X_v des couleurs des arêtes incidentes à v qu'on pourra représenter avec le k -uplet $\{x_1, x_2, \dots, x_k\}$, où x_i est le nombre d'arêtes incidentes à v possédant la couleur i . On souhaite colorer les arêtes de façon à ce que

deux sommets voisins ne possèdent pas le même multi-ensemble (vecteur à K -composantes).

La Figure 3.11 donne un exemple de coloration par multi-ensembles pour le graphe de Petersen. Pour chaque sommet, on indique son k -uplet.

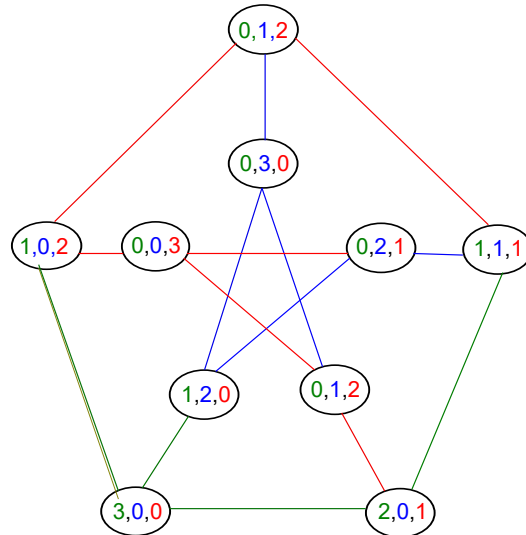


FIGURE 3.11 – Exemple de coloration par multi-ensembles.

Cette coloration a été définie en 2005 par Aldred, Addario-Berry, Reed et Dalal qui ont émis la conjecture suivante :

Conjecture 2 [19].

Il est possible de colorier par multi-ensembles tout graphe ne possédant pas d'arête isolée en utilisant trois couleurs.

Le graphe donné dans la Figure 3.11 nécessite trois couleurs. Cela montre en particulier que le graphe de Petersen vérifie la conjecture 2.

3.3.6.1 Décomposition en sous-graphes localement irréguliers

Un graphe localement irrégulier est un graphe dans lequel pour toute paire $\{u, v\}$ de sommets adjacents, dans G ont des degrés différents ($d_u \neq d_v$).

La décomposition d'un graphe en sous-graphes localement irréguliers consiste à colorier les arêtes du graphe de façon impropre (Coloration impropre des arêtes qui permette de distinguer tous les sommets via une fonction de codage c telle que

$$c(v) = \bigcup_{v \in e} f(e)$$

), de telle façon que dans chaque graphe partiel induit par les arêtes d'une même couleur, deux sommets voisins aient toujours des degrés différents de 1 et 2.

La Figure 3.12 donne un exemple de décomposition en sous-graphes localement irréguliers pour le graphe de Petersen.

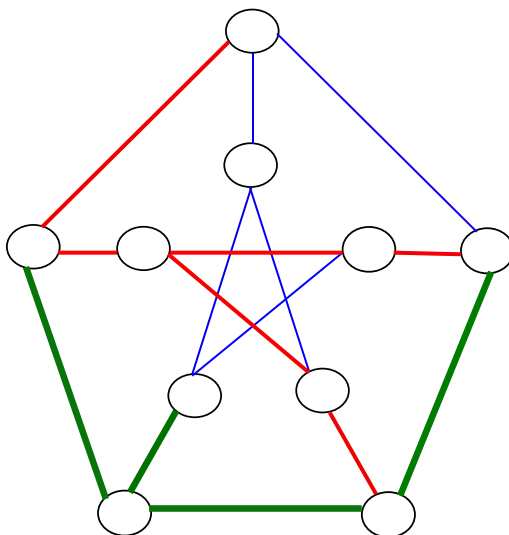


FIGURE 3.12 – Exemple de décomposition en sous-graphes localement irréguliers.

Tous les graphes ne sont pas décomposables de cette façon. Dans [24], il est montré que les graphes non décomposables en sous-graphes localement irréguliers sont les suivants :

- les cycles ayant un nombre impair de sommets ;
- les chaînes ayant un nombre pair de sommets ;
- la famille ζ (Famille de graphes comprenant tous les graphes indécomposables en sous-graphes localement irréguliers à l'exception des cycles d'ordre non divisible par quatre et des chaînes de taille impaire) définie de la façon suivante :
 - le graphe complet de trois sommets K_3 appartient à ζ ;
 - soit $G \in T$ contenant un triangle possédant au moins un sommet v de degré 2. Alors le graphe obtenu en reliant à v , soit une chaîne de longueur paire, soit une chaîne de longueur impaire dont l'autre extrémité est collée à un triangle, appartient à ζ .

La Figure 3.13 montre Un exemple de graphe appartenant à la famille ζ Tous les autres graphes

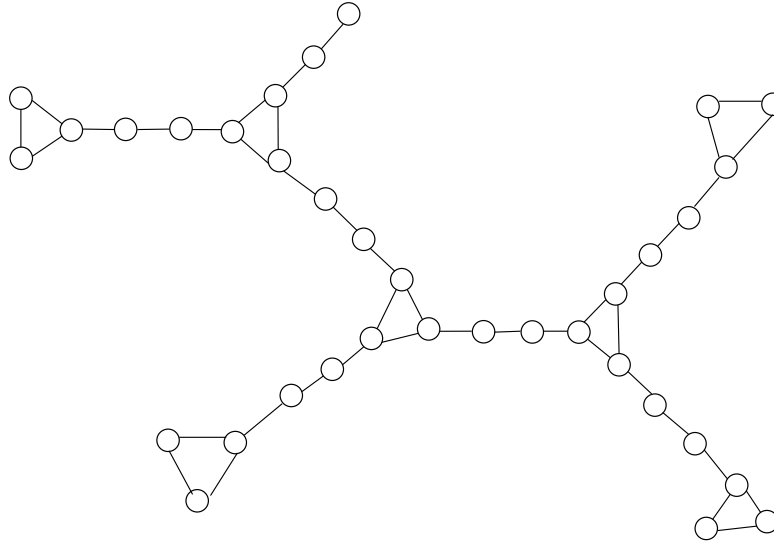


FIGURE 3.13 – Exemple d'un graphe non décomposable

sont décomposables en sous-graphes localement irréguliers.

Ce type de coloration à été défini en 2005 par Baudon, Bensmail, Woźniak et Przybyło qui ont émis la conjecture suivante :

Conjecture 3 [24]

Tout graphe connexe G n'étant ni une chaîne de longueur impaire, ni un cycle de taille impaire, ni un graphe de la famille \mathfrak{S} , est décomposable en trois sous-graphes localement irréguliers.

Le graphe donné en exemple dans la Figure 3.12 nécessite trois couleurs. Cela montre en particulier que si la Conjecture 3 est vérifiée.

3.3.7 Coloration par multi-sommes

Le but de cette partie est d'étudier une nouvelle forme de coloration de graphes, appelée coloration par multi-sommes. La coloration par multi-sommes est une généralisation de trois coloration par somme, coloration par multi ensemble et la décomposition en sous-graphe localement irréguliers. On attribue une couleur et un poids à chaque arête, puis pour chaque sommet v , on fait la somme des poids des arêtes incidentes \hat{v} de la même couleur pour obtenir un tuple d'entiers désigné comme étant la multi-somme de v , qu'on notera comme un p -uplet d'entiers. La

paire couleur-poids d'une arête est appelée valeur de l'arête. L'ensemble des valeurs pouvant être données à une arête dans une coloration par multi-sommes utilisant p couleurs et q poids est noté ε_p^q .

Plus formellement, pour un graphe G et pour c une coloration par multi-sommes utilisant p couleurs et q poids, on utilisera les notations suivantes :

- l'opérateur multi-somme \oplus est défini de la façon suivante :

$$(a_1, a_2, \dots, a_p) \oplus (b_1, b_2, \dots, b_p) = (a_1 + b_1, a_2 + b_2, \dots, a_p + b_p)$$

- l'opérateur \ominus désigne l'opération opposée de \oplus :

$$(a_1, a_2, \dots, a_p) \ominus (b_1, b_2, \dots, b_p) = (a_1 - b_1, a_2 - b_2, \dots, a_p - b_p)$$

- l'opérateur \otimes désigne le produit d'une multi-somme par un entier :

$$a \otimes (b_1, b_2, \dots, b_p) = (a \times b_1, a \times b_2, \dots, a \times b_p)$$

- pour une arête $e \in E(G)$ possédant la couleur K et le poids m , on représente la valeur de e par le p -uplet

$$c(e) = (e_1 := 0, e_2 := 0, \dots, e_K := m, e_{k+1} := 0, \dots, e_p := 0)$$

- pour un sommet $v \in V(G)$, sa multi-somme $c(v)$ est définie de la façon suivante :
soit $N(v) = \{x_1, x_2, \dots, x_{d(v)}\}$, l'ensemble des sommets voisins de v , alors :

$$c(v) = (a_1, a_2, \dots, a_p) = c(vx_1) \oplus c(vx_2) \oplus \dots \oplus c(vx_{d(v)})$$

- pour un p -uplet $\alpha = (a_1, a_2, \dots, a_p)$, α_i désigne sa i -ème composante.

On utilise la même notation pour désigner la valeur d'une arête et la multi-somme d'un sommet afin d'unifier les notations pour les opérandes des opérateurs \oplus et \ominus . En effet ces opérateurs peuvent être utilisés en utilisant comme opérandes des multi-sommes et des valeurs d'arêtes.

Trois variantes de coloration par multi-sommes avec différentes contraintes sur les multi-sommes des sommets voisins vont être étudiées :

1. **la coloration faible par multi-sommes** : dans ce type de coloration, il faut que pour deux sommets voisins, la multi-somme soit différente, c'est-à-dire qu'il y ait au moins une couleur pour laquelle la somme des poids des arêtes incidentes soit différente pour les deux

sommets. De façon plus formelle on a :

$$\forall xy \in E(G), \exists i \in \{1, \dots, p\}, c_i(x) \neq c_i(y)$$

La Figure 3.14 montre un exemple de coloration faible par multi-sommes utilisant deux poids et deux couleurs. Pour chaque arête e , on indique sur le dessin sa valeur $c(e)$ et pour chaque sommet v , on indique sur le dessin sa multi-somme $c(v)$.

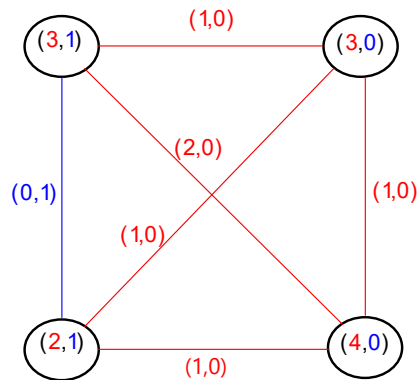


FIGURE 3.14 – Exemple de coloration faible par multi-sommes.

2. **la coloration standard par multi-sommes** : dans ce type de coloration, il faut que pour deux sommets voisins, la composante de la multi-somme associée à la couleur de l'arête les reliant soit différente. De façon plus formelle on a :

$$\forall xy \in E(G), \exists i \in \{1, \dots, p\}, c_i(xy) \neq 0 \wedge c_i(x) \neq c_i(y)$$

i.e :

$$C_1(x) \neq C_1(y) \text{ et } C_1(xy) \neq 0$$

ou

$$C_2(x) \neq C_2(y) \text{ et } C_2(xy) \neq 0$$

La Figure 3.15 montre un exemple de coloration standard par multi-sommes utilisant deux poids et deux couleurs. Pour chaque arête e , on indique sur le dessin sa valeur $c(e)$ et pour chaque sommet v , on indique sur le dessin sa multi-somme $c(v)$. On remarque que chaque sous-graphe induit par une couleur est colorié par sommes.

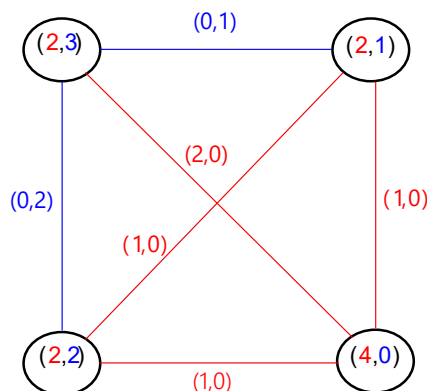


FIGURE 3.15 – Exemple de coloration standard par multi-sommes.

3. **La coloration forte par multi-sommes** : dans ce type de coloration, il faut que pour deux sommets voisins, chacune des composantes de leurs multi-sommes soit nulle ou différente pour les deux sommets. De façon plus formelle on a :

$$\forall xy \in E(G), \exists i \in \{1, \dots, p\}, c_i(x) \neq c_i(y) \vee (c_i(x) = 0 \wedge c_i(y))$$

i.e :

Soit

$$C_1(x) \neq C_1(y) \text{ et } C_2(x) \neq C_2(y) \neq 0$$

ou bien

$$C_1(x) = C_1(y) = 0 \text{ et } C_2(x) \neq C_2(y)$$

$$C_2(x) = C_2(y) = 0 \text{ et } C_1(x) \neq C_1(y)$$

La Figure 3.16 montre un exemple de coloration forte par multi-sommes utilisant deux poids et deux couleurs. Pour chaque arête e , on indique sur le dessin sa valeur $c(e)$ et pour chaque sommet v , on indique sur le dessin sa multi-somme $c(v)$.

Il est possible de s'intéresser à une coloration par multi-sommes possédant une contrainte encore plus forte, où l'on aurait voulu des composantes différentes pour toutes les couleurs pour chaque paire de sommets voisins (en interdisant une composante nulle pour les deux sommets en même temps). Cependant ce type de coloration aurait été trop contraignant dans la mesure où un graphe colorier avec p couleurs et q poids dans une telle coloration, n'est pas forcément coloriable avec $p + 1$ couleurs et q poids. Un graphe ne pourrait utiliser au maximum que

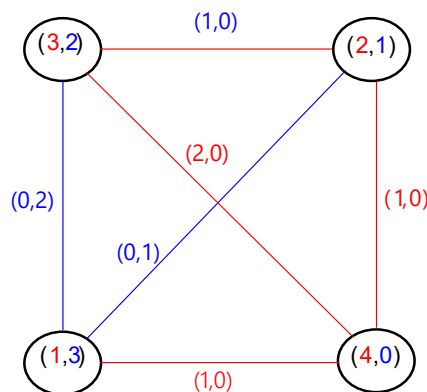
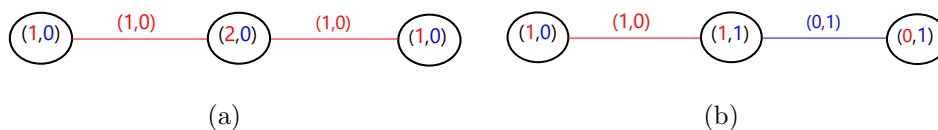


FIGURE 3.16 – Exemple de coloration forte par multi-sommes.

$\min\{\max(d(u), d(v)) | uv \in E(G)\}$ couleurs (ce paramètre est appelé le degré minmax).

Par exemple, pour la chaîne P_3 de trois sommets, il est possible de colorier P_3 avec une couleur et un poids mais impossible de le colorier avec plus d'une couleur.

la Figure 3.17 montre les deux façons possibles de colorier P_3 avec deux couleurs et un poids : dans (a), la composante 2 vaut 0 pour tous les sommets et dans (b), le sommet du milieu possède la même composante 1.

FIGURE 3.17 – Deux façons de colorier P_3 avec deux couleurs et un poids.

On cherchera dans toutes les variantes considérées à minimiser le nombre de poids et de couleurs utilisés. Un graphe est dit (faiblement / de façon standard / fortement) (p, q) -coloriable par multi-sommes s'il existe une coloration (faible / standard / forte) par multi-sommes utilisant p couleurs et q poids. Un graphe est dit (faiblement / de façon standard / fortement) (p, q) -chromatique par multi-sommes s'il est (faiblement / de façon standard / fortement) (p, q) -coloriable par multi-sommes et s'il n'existe pas de (faiblement / de façon standard / fortement) $(p - 1, q)$ -coloration ni de $(p, q - 1)$ -coloration. Une (p, q) -coloration (faible / standard / forte) d'un graphe (faiblement / de façon standard / fortement) (p, q) -chromatique est dite optimale. Un même graphe peut donc avoir plusieurs colorations optimales avec un nombre de poids et de couleurs différents. Une coloration pour laquelle toutes les arêtes n'ont pas une valeur est dite partielle.

La Figure 3.18 montre le $(2, 1)$ -chromatique de C_4 .

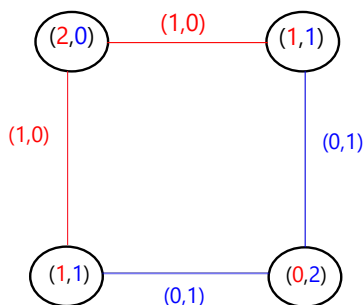


FIGURE 3.18 – C_4 est fortement $(2, 1)$ -chromatique .

Comme dans une coloration forte, les sommets voisins ont les composantes de la couleur qui les relie différentes et que dans une coloration standard, les voisins ont au moins une composante différente, on peut faire la remarque suivante :

Remarque Une (p, q) -coloration forte par multi-sommes est une (p, q) -coloration standard par multi-sommes et une (p, q) -coloration standard par multi-sommes est une (p, q) -coloration faible par multi-sommes.

Ainsi si l'on montre qu'un graphe admet une (p, q) -coloration forte (respectivement standard) par multi-sommes alors ce graphe admet aussi une (p, q) -coloration standard (respectivement faible) par multi-sommes. De même si l'on montre qu'un graphe n'admet pas une (p, q) -coloration faible (respectivement standard) par multi-sommes alors ce graphe n'admet pas une (p, q) -coloration standard (respectivement forte) par multi-sommes.

Cependant si l'on montre qu'un graphe est fortement (p, q) -chromatique (respectivement (p, q) -chromatique de façon standard) par multi-sommes, cela ne suffit pas à dire qu'il est (p, q) -chromatique de façon standard (respectivement faiblement (p, q) -chromatique) par multi-sommes. En effet comme la variante de coloration par multi-sommes est moins contraignante dans le deuxième cas, il est possible que le graphe nécessite un poids ou une couleur de moins dans la deuxième variante.

3.3.7.1 Liens entre les différentes colorations

Une coloration (faible, standard ou forte) par multi-sommes n'utilisant qu'une seule couleur est équivalente à une coloration par sommes. De même une coloration standard par multi-somme n'utilisant qu'un seul poids est équivalente à une décomposition en sous graphes localement irréguliers. Une coloration faible par multi-sommes n'utilisant qu'un seul poids quant à elle correspond à une coloration par multi-ensembles. Les résultats obtenus lors des études de la coloration par sommes, de la coloration par multi-ensembles et de la décomposition en sous-graphes localement irréguliers peuvent donc être directement convertis en terme de coloration par multi-sommes.

La proposition suivante établit un lien entre la coloration par sommes et la coloration par multi-ensembles :

Proposition [7] Soit G un graphe connexe différent de K_2 et k un entier naturel non nul. Si G est fortement $(1, k)$ -coloriable par multi-sommes, alors G est faiblement $(k, 1)$ -coloriable par multi-sommes.

Démonstration : Soit G un graphe connexe différent de K_2 et c une $(1, k)$ -coloration forte par multi-sommes de G . Construisons une $(k, 1)$ -coloration faible par multi-sommes c' de G . Donnons aux arêtes de G la couleur i quand celles-ci possèdent le poids i dans la coloration c et donnons le poids 1 à toutes les arêtes. Pour toute paire de sommets voisins x et y , on a :

$$c_1(x) = w_1(x) + w_2(x) \times 2 + \dots + w_p(x) \times p$$

et

$$c_1(y) = w_1(y) + w_2(y) \times 2 + \dots + w_p(y) \times p$$

les multi-sommes de respectivement x et y dans la coloration c où $w_i(v)$ est le nombre d'arêtes incidentes à v possédant le poids i . Comme $c_1(x) \neq c_1(y)$, il existe un poids i tel que $w_i(x) \neq w_i(y)$. Dans la nouvelle coloration par multi-sommes, on a $c'(x) = (w_1(x), w_2(x), \dots, w_p(x))$ et $c'(y) = (w_1(y), w_2(y), \dots, w_p(y))$, donc il y a une composante différente pour ces deux sommets.

3.4 Les méthodes de coloration simple des graphes

3.4.1 Algorithme de coloration de WELSH et POWELL [6]

Soit $G = (V, E)$ un graphe simple et connexe :

Étape 1 :

- Ordonner les sommets selon l'ordre décroissant de leur degré ;
- Donner à chaque sommet, son numéro d'ordre dans la liste obtenue ;

Étape 2 :

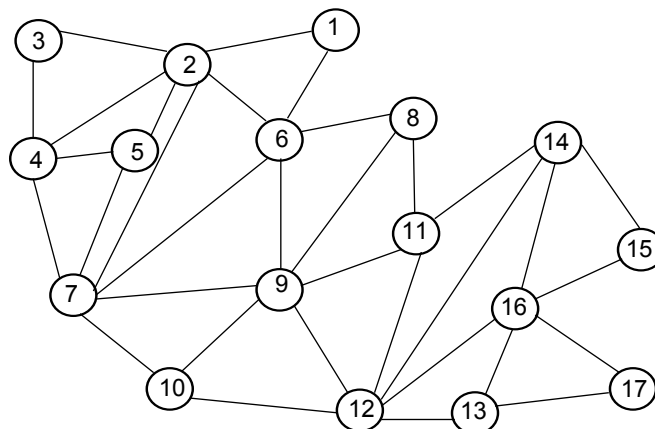
- Parcourir la liste dans l'ordre en attribuant une couleur non encore utilisée, au premier sommet non encore colorié ;
- Attribuer cette même couleur en suivant la liste à chaque sommet non encore colorié et non adjacent à un sommet de cette couleur ;

Étape 3 :

- Revenir à l'étape 2, tant qu'il reste des sommets non coloriés ;
- Sinon s'arrêter : la coloration du graphe est terminée.

Exemple :

Étant donné un graphe planaire (voire la figure) de 17 sommets, donc la coloration de type de graphe fournit une coloration de graphe utilise au plus 4 couleur.



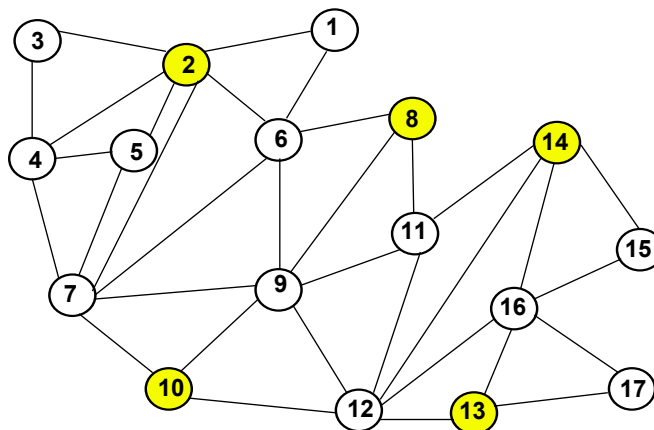
En utilise l'algorithme de Walsh and Powelle

1. **Itération 0** :Premièrement on va numéroter les sommets de notre graphe

Classer les sommet du graphe dans un ordre décroissant de leur degré en attribuant à chacun des sommets un numéro d'ordre dans la liste obtenue. Les sommets classer comme suit : $v_2, v_7, v_9, v_{12}, v_6, v_{16}, v_4, v_{11}, v_{14}, v_5, v_8, v_{10}, v_{13}, v_1, v_3, v_{17}, v_{15}$

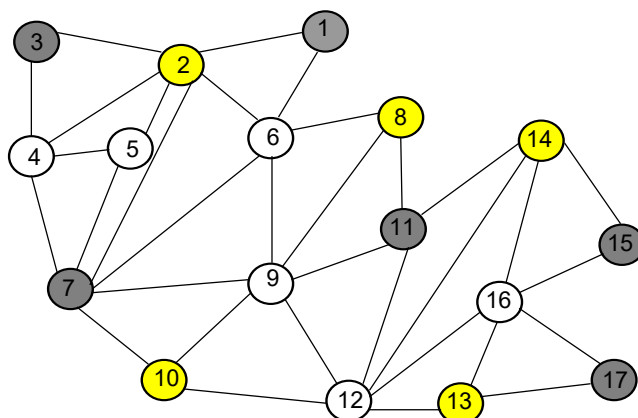
2. **Itération 1** :On commence à colorier le sommet v_2 qui le plus haut degré avec la couleur jaune .

Et on affecte cette couleur aux sommets $v_8, v_{10}, v_{13}, v_{14}$ qui ne sont pas adjacente à v_2 . voici La Figure

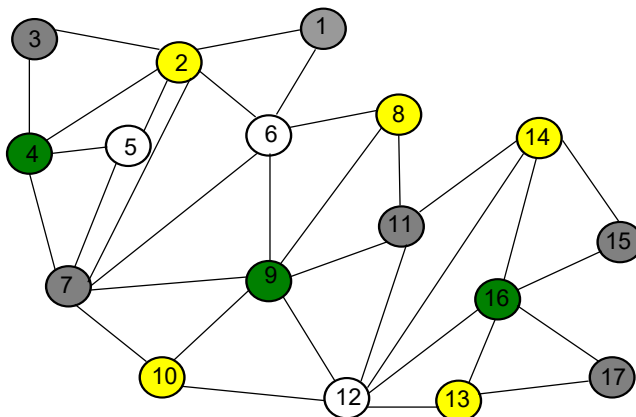


3. **Itération 2** :On affecte la couleur gris au sommet v_7 ayant le plus haut degré.

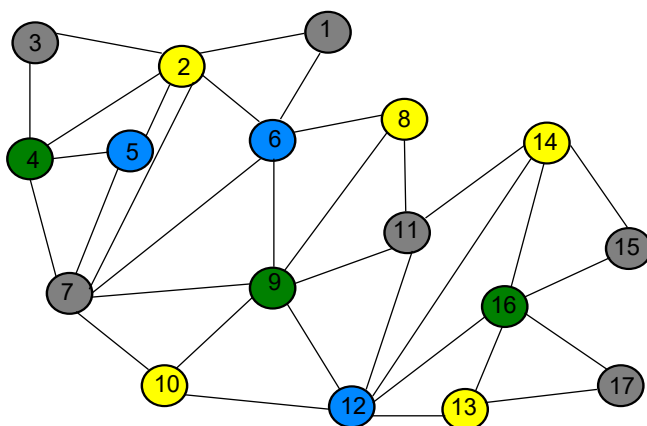
Et on affecte cette couleur aux sommets $v_1, v_3, v_{11}, v_{15}, v_{17}$ qui non encore colorier et ne sont pas adjacents à v_7 .



4. **Itération 3** : On affecte la couleur vert au sommet v_9 ayant le plus haut degré. Et on affecte cette couleur aux sommets v_4, v_{16} qui ne sont pas adjacents à v_9 . voici La Figure.



5. **Itération 4** : On affecte la couleur bleu pour v_5 ayant le plus haut degré parmi les sommets encore coloriée, et aux sommets v_6, v_{12} qui ne sont pas adjacents à v_5 .



Tous les sommets sont colorés donc on arrête .

3.4.2 Algorithme DSATUR [25]

On considère un graphe $G = (V; E)$ simple et connexe. Pour chaque sommet v de V , on calcule le degré de saturation $\mathbf{DSAT}(v)$ de la manière suivante :

Si aucun voisin de v n'est colorié alors

$$\mathbf{DSAT}(v) = \text{degré}(v)$$

Sinon

$\mathbf{DSAT}(v)$ = le nombre de couleurs différentes utilisées dans le premier voisinage de v .

L'algorithme **DSATUR** est un algorithme de coloration séquentiel, au sens où il colorie un seul sommet à la fois et tel que :

Au départ le graphe n'est pas colorié.

On colorie un sommet non déjà colorié.

On stoppe **DSATUR** quand tous les sommets de G sont coloriés.

Dans le détail l'algorithme est le suivant :

1. Ordonner les sommets par ordre décroissant de degré.
2. Colorier un sommet de degré maximum avec la couleur 1.
3. Choisir un sommet non colorié avec \mathbf{DSAT} maximum. Si conflit, choisir celui avec degré maximum.
4. Colorier ce sommet par la plus petite couleur possible.
5. Si tous les sommets sont coloriés alors on arrête.

Sinon aller en 3.

Exemple

Étant donné un réseau radio qui contient huit émetteurs reliés entre eux par des interférences, pour cela il faut pouvoir allouer différentes fréquences aux émetteurs, de telle sorte que deux émetteurs voisins ne soient pas de même fréquence.

pour résoudre notre problème, nous devons représenter le réseau d'émetteurs par un graphe, telle que chaque sommet représente un émetteur et chaque arête représente une interférence.

On obtient le graphe dans la Figure 3.19 :

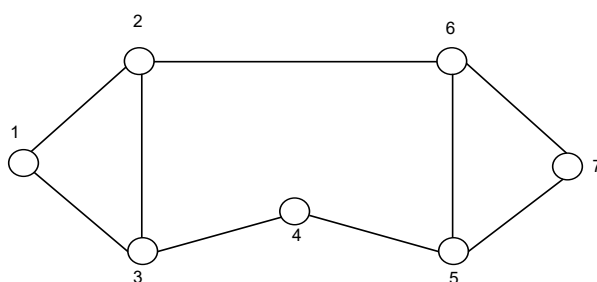


FIGURE 3.19 – Un Graphe 1.

1. **Itération 0** : On ordonne les sommets par ordre décroissant de degré on aura le tableau suivant :

Sommet	v_2	v_3	v_5	v_6	v_1	v_4	v_7
Degré	3	3	3	2	2	2	2

2. **Itération 1** : maximum, mais tous les sommets ont le même degré de saturation, alors on choisit celui de degré maximum (v_3 ou bien v_6), soit v_3 et lui donne la couleur noire, alors $DSAT(v_4) = DSAT(v_1) = 2$, et pour les autres sommets, leur **DSAT** est égal à 1.
3. **Itération 2** : On choisit un autre sommet de **DSAT** maximum (v_1, v_4), soit v_1 , on lui affecte la couleur rouge, le **DSAT** des autres sommets ne vont pas changer.
4. **Itération 3** : On choisit un sommet de **DSAT** maximum (v_4), on lui donne la couleur rouge (v_4 n'est pas adjacent à v_1), ainsi les **DSAT** des autres sommets ne vont pas changer.

5. **Itération 4** : Il nous reste deux sommets non encore coloriés (v_6, v_7), avec $\text{DSAT}(v_6) = \text{DSAT}(v_7) = 1$, donc on choisit le sommet de degré maximum (v_6), on lui affecte la couleur noire (car v_6 n'est pas adjacent à v_3).

Il nous reste le sommet v_7 et on lui donne la couleur rouge.

La Figure 3.20 représente le graphe obtenu.

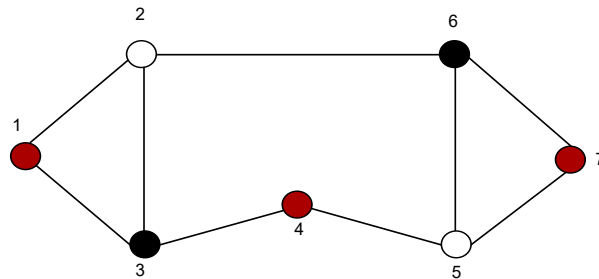


FIGURE 3.20 – Un Graphe $K_{2,3}$.

3.4.3 Algorithme de Coloration à jeton [11]

La coloration se fait par circulation d'un jeton, c'est-à-dire qu'à un instant donné, un seul sommet du graphe est en train de choisir une couleur.

La circulation du jeton se fera sur l'arbre recouvrant exclusivement. Cet arbre se construira au mesure de la coloration, par l'envoi ou non de la couleur à ses voisins. Dans le graphe, un sommet aura des voisins, un père, des aïeux, et des fils.

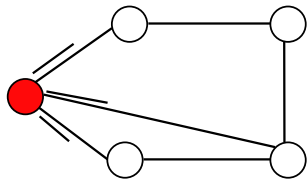
Père : le sommet qui lui a demandé de se colorier.

Aïeux : tous les voisins coloriés (ils se sont coloriés avant moi).

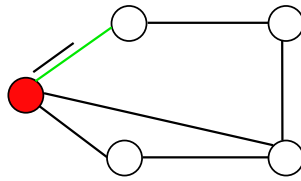
Fils : tous mes voisins non coloriés.

Un sommet donné ne communiquera sa couleur qu'à ses fils. Ses aïeux, qui sont prioritaires sur lui pour choisir leur couleur n'ont pas besoin de connaître.

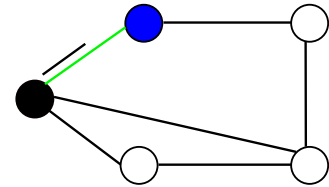
Exemple



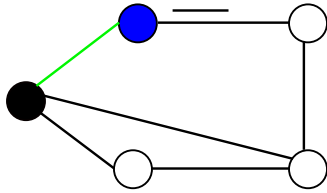
(a) **Phase 1** : L'élue choisit sa couleur et en informe ses voisins.



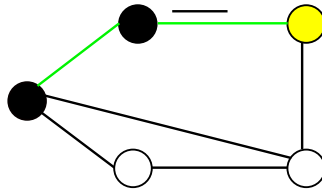
(b) **Phase 2** : L'élue demande à un de ses fils de se colorer.



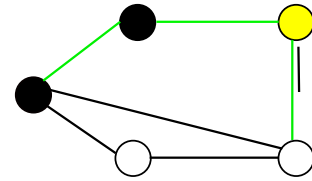
(c) **Phase 3** : Le sommet choisit sa couleur et en informe ses fils.



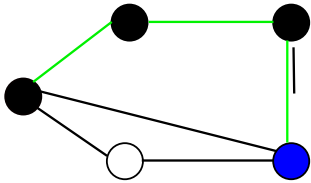
(d) **Phase 4** : Le sommet demande à un de ses fils de se colorer.



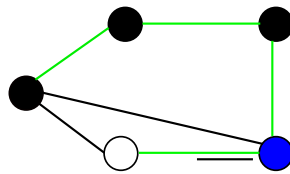
(e) **Phase 5** : Le sommet choisit sa couleur et en informe ses fils



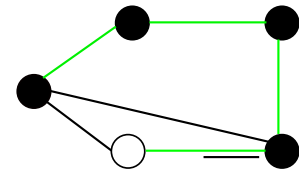
(f) **Phase 6** : Le sommet demande à un de ses fils de se colorer.



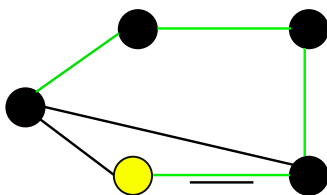
(g) **Phase 7** : Le sommet choisit sa couleur et en informe ses fils.



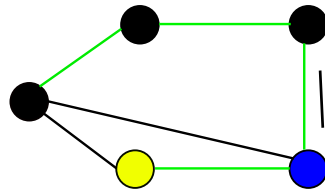
(h) **Phase 8** : Le sommet demande à un de ses fils de se colorer.



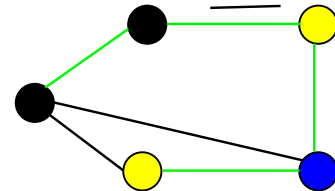
(i) **Phase 9** : Le sommet choisit sa couleur et en informe ses fils



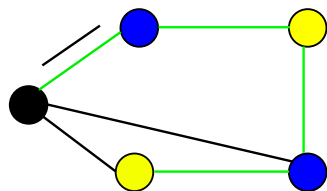
(j) **Phase 10** : Le sommet choisit sa couleur Il n'a plus de fils : remonte OK



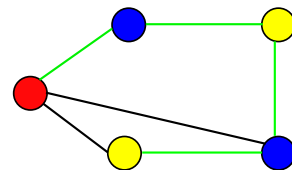
(k) **Phase 11** : Plus de fils non coloré. Remonte OK



(l) **Phase 12** : Plus de fils non coloré. Remonte OK



(m) **Phase 13** : Plus de fils non coloré. Remonte OK.



(n) **Phase 14** : Plus de fils non coloré. ⇒ Coloration réussie.

Nouvelle classe de graphe coloriable par sommes et Applications

4.1 Introduction

Dans cette partie, nous allons présenter une classe de graphe vérifiant la conjecture de Karonski (2004)[20] et on a programmé l'algorithme de WELSH end POWELL que nous avons appliqué pour minimiser les interférences dans le problème d'affectation de fréquences, et pour déterminer le nombre chromatique de la carte géographique des daïras de la wilaya de Annaba.

4.2 Coloration par sommes sur les arêtes d'une classe de graphe

Définitions 4.2.1 Le graphe H_n $n \geq 1$ est donné par la somme cartésienne de la chaîne de la longueur 1 (P_1) et de la chaîne de la largeur n (P_n).

- H_1 est donné par la Figure 4.1

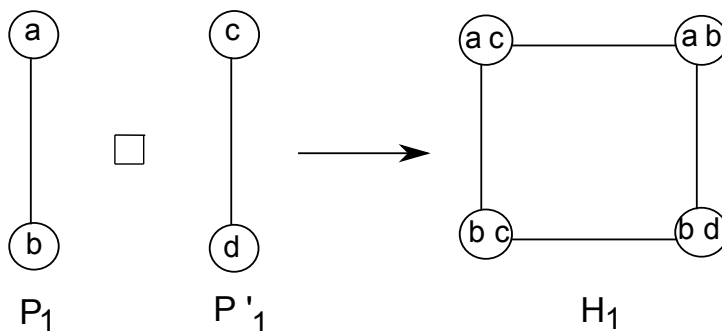


FIGURE 4.1 – Graphe H_1

- H_2 est donné par la figure 4.2

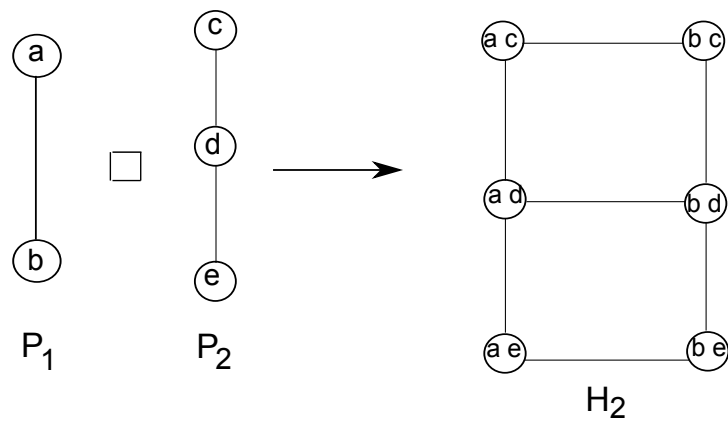


FIGURE 4.2 – Graphe H_2

Théorème : Pour tout $n \geq 1$, H_n peut être colorier par sommes sur les arêtes en utilisant seulement les poids appartenant à l'ensemble $\{1, 2\}$.

Preuve :

1. Il est clair que H_1 est colorié par sommes sur les arêtes en utilisant les valeurs de l'ensemble $\{1, 2\}$. Comme le montre la Figure 4.3

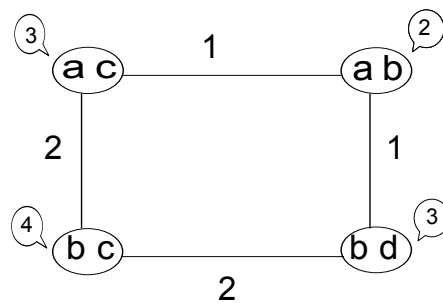


FIGURE 4.3 – Coloration par sommes sur les arêtes de H_1

2. La coloration par sommes sur les arêtes de H_2 est donnée par la Figure 4.4

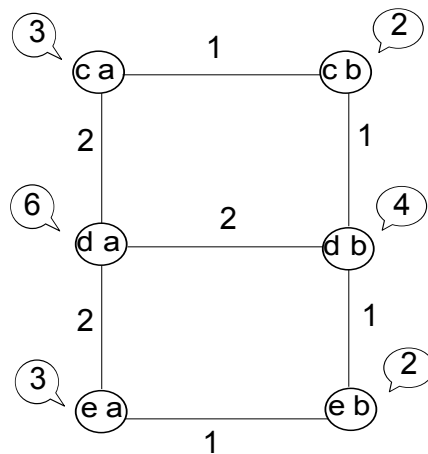


FIGURE 4.4 – Coloration par sommes sur les arêtes de H_2

3. Pour $n \geq 3$, on distingue les deux cas suivants :

✓ n est **impair**, $H_n = H_{2p-1}$ $p \geq 2$, H_{2p-1} est obtenu à partir de deux copies, disjointes H'_{p-1} et H''_{p-1} de H_{p-1} .

Telle que pour deux sommets x et y adjacents de degré 2 de H'_{p-1} et deux sommets x' , y' adjacents de degré 2 de H''_{p-1} , x est relié à x' par une arête et y est relié à y' par une autre arête comme le montre la Figure 4.5

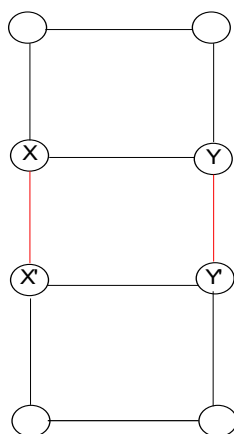
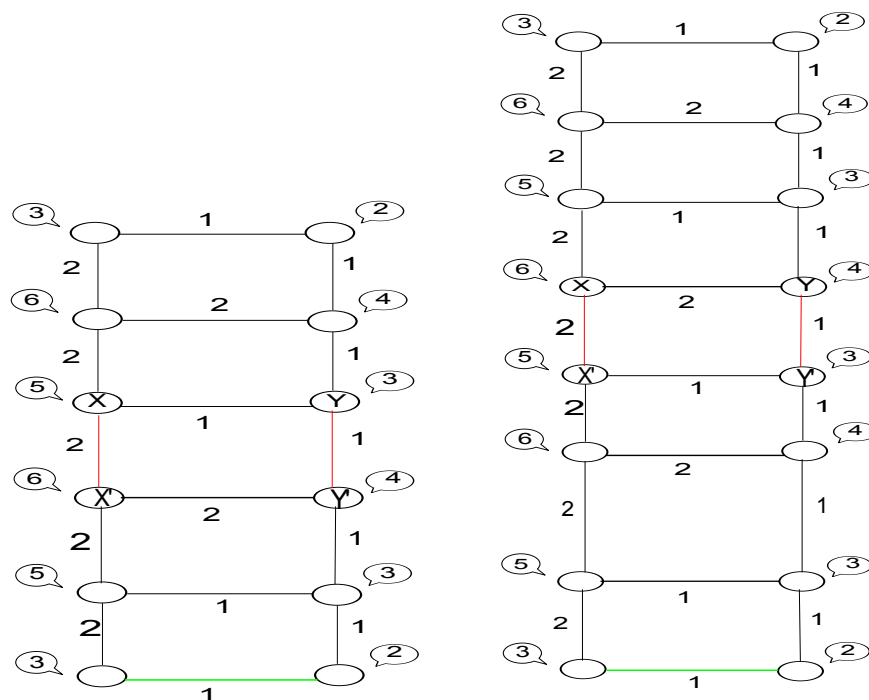


FIGURE 4.5 – Construction de H_3 à partir de deux copies disjointes de H_1

Pour avoir une coloration par sommes sur les arêtes de H_{2p-1} il faut colorier y, y' par 1, x, x' par 2 et les arêtes horizontales de H_{2p-1} par les valeurs 1 et 2 d'une façon alternée. En attribuant 1 à première (en tenant compte de l'ordre des arêtes de haut en bas) et remplacer le poids 2 de la dernière arêtes par le poids 1.

La Figure 4.6 montre la coloration par sommes sur les arêtes de H_5 et H_7



(a) H_5

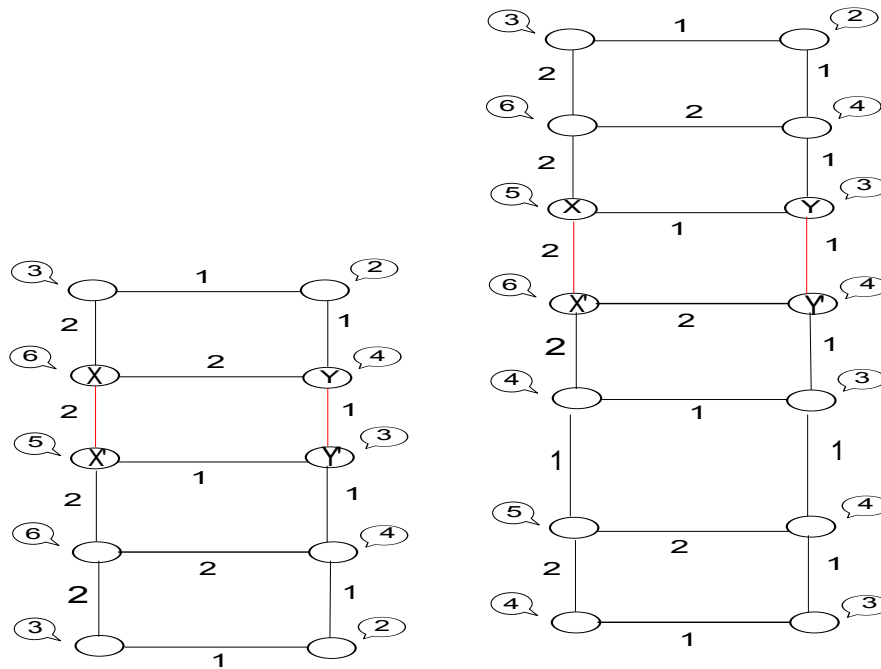
(b) H_7

FIGURE 4.6 – Coloration par sommes de H_5 et H_7

✓ n est **pair**, H_n avec $H_n = H_{2p}$ $p \geq 2$. H_{2p} est obtenu à partir d'une copie H'_{p-1} de H_{p-1} et une copies H'_p de H_p , tels que pour deux sommets x, y adjacents de degré 2 de H'_{p-1} et deux sommets x', y' adjacents de H'_p , x est relié à x' par une arêtes et y est relié à y' par une autre.

Pour avoir une coloration par sommes sur les arêtes de H_{2p} , il faut colorier y, y' par le poids 1, x, x' par 2 et les arêtes horizontale de H_{2p} par les valeurs 1 et 2 d'une façon alternée en attribuant 1 à la première (en commençant de haut en bas).

La Figure 4.7 donne la coloration par sommes sur les arêtes de H_4 et H_6



(a) H_4

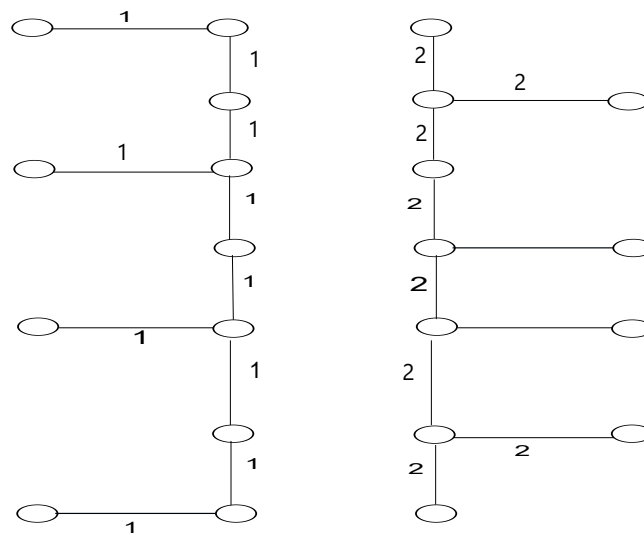
(b) H_6

FIGURE 4.7 – Colaration par sommes de H_4 et H_6

La classe que nous avons défini vérifie la conjecture de Karonski (2004)[20].

Le graphe H_n est décomposable en sous-graphes localement irréguliers, comme le montre la Figure 4.8 et 4.9 pour les graphe H_6 et H_7 .

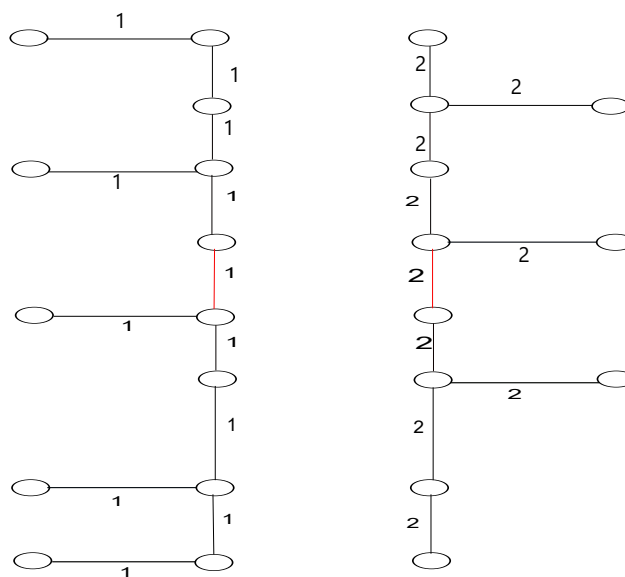
Pour avoir cette décomposition nous allons répartir les arêtes de H_n suivant leur poids. Les arêtes marquées par 1 donne le premier arbre et les arêtes marquées par 2 donne le deuxieme arbre.



(a) Arbre 1

(b) Arbre 2

FIGURE 4.8 – La Décomposable en sous-graphes localement irréguliers de H_6



(a) Arbre 1

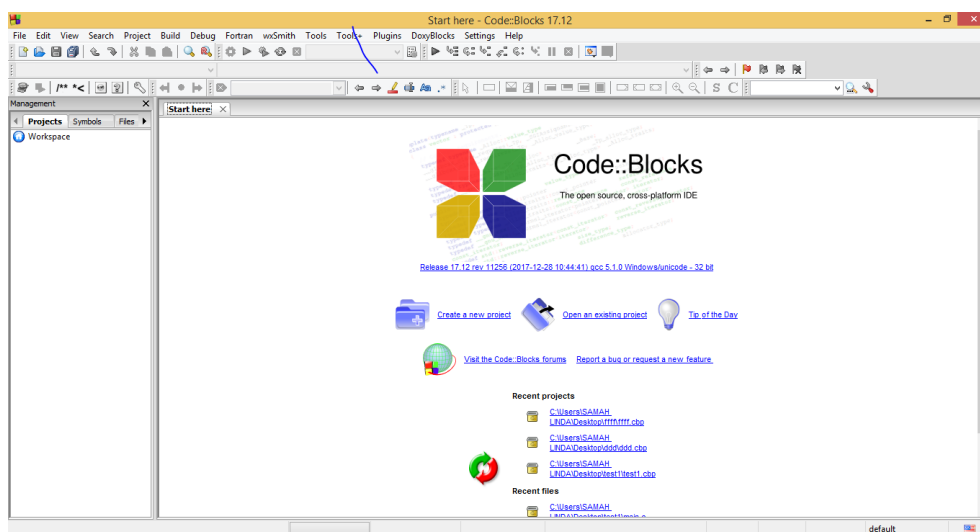
(b) Arbre 2

FIGURE 4.9 – La Décomposable en sous-graphes localement irréguliers de H_7 .

Remarque Pour avoir deux arbres de même nombre d'arêtes nous allons attribuer le poids 2 à la première arête (cas où $n = 2p - 1$)

4.3 Introduction de CodeBlocks

Code Blocks est un environnement de développement intégré libre et multiplateforme. Il est écrit en C++ grâce à la bibliothèque wxWidgets. Code Blocks est orienté C++, mais il supporte d'autres langages comme FORTRAN développé pour Linux, Windows et Mac OS X. Des utilisateurs indiquent avoir réussi à compiler le code source sous FreeBSD. L'illustration ci-dessous montre l'apparence de la fenêtre de l'interface utilisateur de Code-Blocks.



Dans Code Blocks, les sources et les paramètres d'un processus de génération sont stockés dans un fichier projet $\ll name.cbp \gg$. Les sources en C/C++ et les fichiers d'entêtes correspondants (ou headers) sont les composants typiques d'un projet. La façon la plus simple de créer un projet est de passer par la commande "Fichier" → "Projet" et de choisir un assistant. Vous pouvez alors ajouter des fichiers au projet via le menu de contexte 'Ajouter des fichiers' de la fenêtre de gestion.

Code Blocks place automatiquement tous les fichiers du projet dans un répertoire qui porte le nom du projet. Pour construire et exécuter un projet, vous pouvez éditer le fichier main. En allant à gauche dans la fenêtre Management dans Source → main.c. Ce fichier comporte la fonction main, fonction principale du programme.

4.4 Résolution du problème d'allocation de fréquences

4.4.1 Concept cellulaire

Une cellule représente l'ensemble des points du territoire couvert par une même station (BTS). Chaque station de base peut posséder plusieurs antennes donnant ainsi naissance à plusieurs cellules (appelées secteurs dans ce cas), on distingue généralement trois cellules. Des stations de base mono-sectorielles, couvrant la zone à 360° , sont utilisées dans les zones très peu peuplées et dans les centres villes pour créer des microcellules. Des stations de base bi-sectorielles, donnant naissance à deux cellules de 180° chacune, sont souvent mises en place aux abords des autoroutes.

Les stations de base tri - sectorielles sont les plus répandues et les plus utilisées pour le réseau Wimax; elles génèrent trois cellules de 120° .

La Figure 4.10 montre un modèle d'un concept cellulaire.

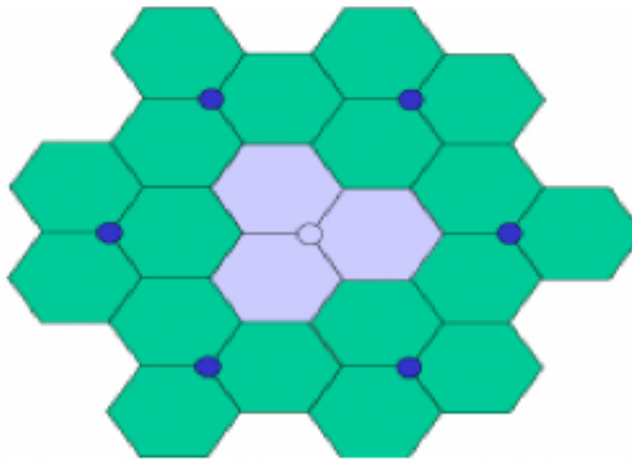


FIGURE 4.10 – Modèle du concept cellulaire.

La forme hexagonale a été universellement adoptée comme représentation théorique du design cellulaire [Mac Donald, 1979]. En effet l'hexagone désigne la forme géométrique la plus proche du cercle (voir la figure 4.9) qui permet un pavage régulier du plan en utilisant le moins de cellules. De plus il garantit une uniformité des distances entre les émetteurs, la régularité des schémas d'antennes et de la propagation des ondes radio en espace libre. La réalité, s'écarte de cette vue théorique. La non régularité des reliefs géographiques (montagnes, plateaux...) et architecturaux (bâtiments, maisons...) fait que la propagation des ondes ne s'effectue pas de la même façon dans toutes les directions. De ce fait, des prolongements, des rétractions voir même des discontinuités importantes apparaissent dans la couverture des cellules.

La Figure 4.11 représente les couvertures cellulères

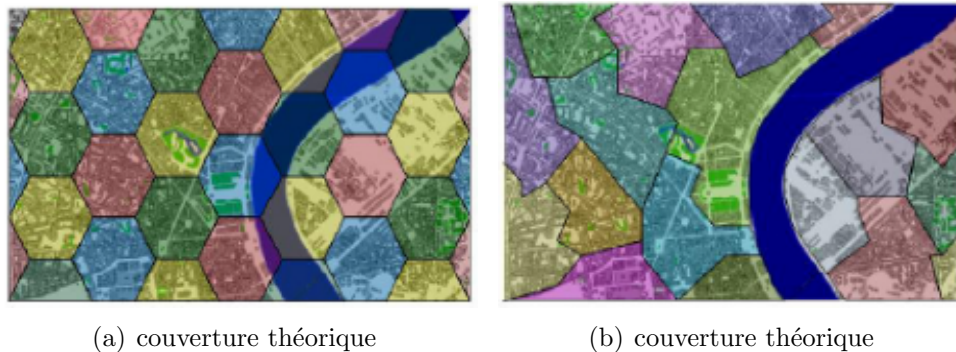


FIGURE 4.11 –

Fonctions des réseaux cellulaire

- Assurer une couverture,
- Assurer une capacité grâce à la réutilisation des ressources fréquentielles,
- Permettre une localisation automatique des stations mobiles,
- Permettre le hand-over (ou hand-off) : « transfert automatique intercellulaire » en cours de communication.

Dans un réseau de télécommunication, il y a des émetteurs émettant chacun sur une fréquence particulière. Imaginons que l'on veuille construire un réseau d'antennes radios. Il faudra réserver certaines fréquences de communication, que l'on attribuera à nos antennes afin qu'elles puissent communiquer entre elles. Il faut aussi minimiser le nombre de fréquences à réserver, car chaque réservation a un coût. Une première modélisation du problème, en radio AM par exemple, montre que si deux antennes "assez proches" géographiquement émettent dans des fréquences trop voisines l'une de l'autre, leur spectres en fréquences se recouvrent et il devient impossible d'extraire l'information nécessaire du signal reçu.

On modélise ainsi le réseau d'antenne par un graphe, appelé graphe d'interférences, dont les sommets sont les antennes, et les arêtes relient deux antennes qui interfèrent entre elles si et seulement si les deux antennes correspondants sont trop proches, comme l'illustre la figure 4.10. Les fréquences à allouer correspondent à des couleurs (représentées par des entiers) avec les quelles nous colorons les sommets du graphe. La coloration doit être une coloration propre.

La Figure 4.12 représente un réseau d'antenne.

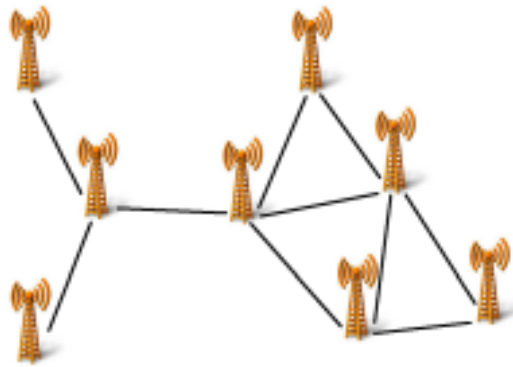


FIGURE 4.12 – Un réseau d'antennes .

4.4.2 Modélisation par la théorie des graphes

Le problème d'affectation de fréquences est un problème de la classe du coloriage de graphe (Graph Coloring Problem).

On attribuer aux antennes reliant des bandes de fréquences pour communiquer avec les usagers.

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

V : L'ensemble des sommets du graphe représentent les antennes.

E : L'ensemble des arêtes du graphe représentent le lien entre des antennes.

Couleurs = représentent les fréquences.

Coloration du graphe = allocation de fréquences sans interférences.

Minimiser le nombre de couleurs = minimiser la largeur de bande nécessaire.

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.

La Figure 4.13 représente un graphe cellulaire

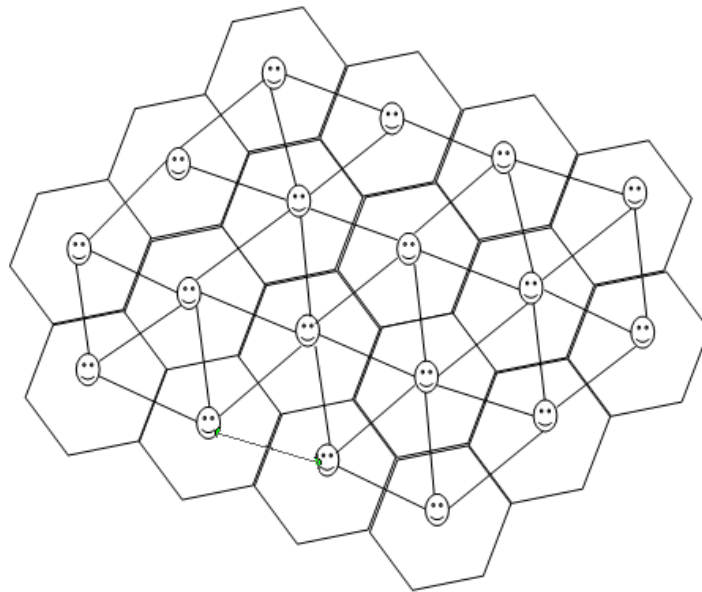
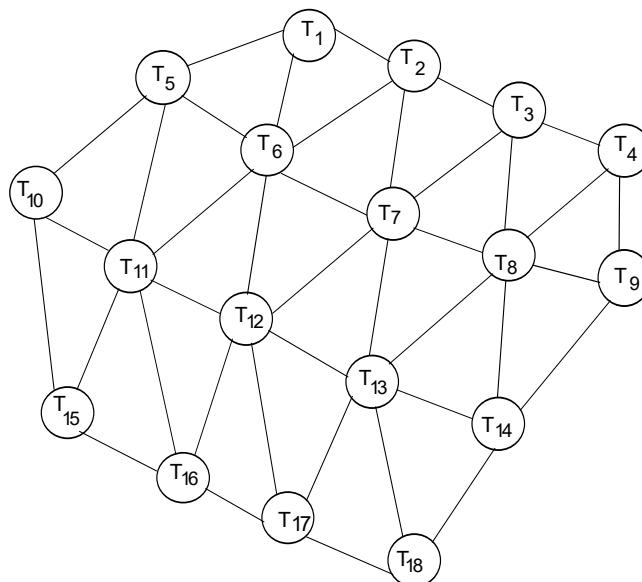


FIGURE 4.13 – Graphe cellulaire

4.4.3 Résolution de problème d'allocation de fréquences

D'après cette modélisation on remarque que les arêtes ne se croisent pas alors nous obtenons un graphe planair suivant.



En utilise l'algorithme de Walsh and Powelle

- **Itération 0** :Premièrement on va numéroter les cellules Classifier les sommet du graphe dans un ordre décroissant de leur degré en attribuant à chacun des sommets un numéro d'ordre dans la liste obtenue. Les sommets sont classer comme suit : $T_6, T_7, T_8, T_{11}, T_{12}, T_{13}, T_2, T_3, T_5, T_{16}, T_{17}, T_{14}, T_1, T_9, T_{10}, T_{15}, T_{18}$.
- **Itération 1** : On commence à colorié le sommet T_6 qui le plus grand degré avec la couleur verte et on affecte cette couleur aux sommets $T_{13}, T_3, T_{16}, T_9, T_{10}$ qui ne sont pas adjacente à T_6 .

Comme le montre le graphe de la Figure 4.14

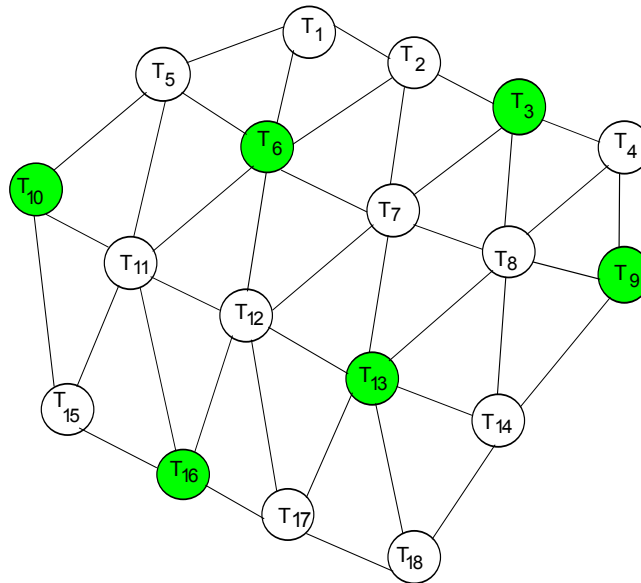


FIGURE 4.14 – Le graphe associé à Itération 1

- **Itération 2** :On affecte la couleur rouge au sommet T_7 ayant le plus grand degré parmi les sommets non coloriés et on affecte cette couleur aux sommets $T_{11}, T_{17}, T_{14}, T_1$ qui ne sont pas adjacents à T_7 .

Comme le montre le graphe de la Figure 4.15

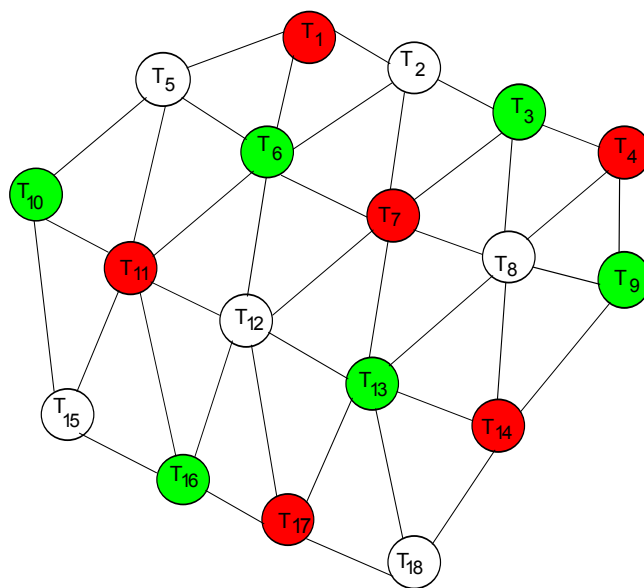


FIGURE 4.15 – Le graphe associé à Itération 2

- **Itération 3** :On affecte la couleur bleue pour T_8 ayant le plus grand degré parmi les sommets non coloriés , et on effect cette couleur aux sommets $T_{12}, T_2, T_5, T_{15}, T_{18}$ qui ne sont pas adjacentes à v_8 .

Comme le montre le graphe de la Figure 4.16

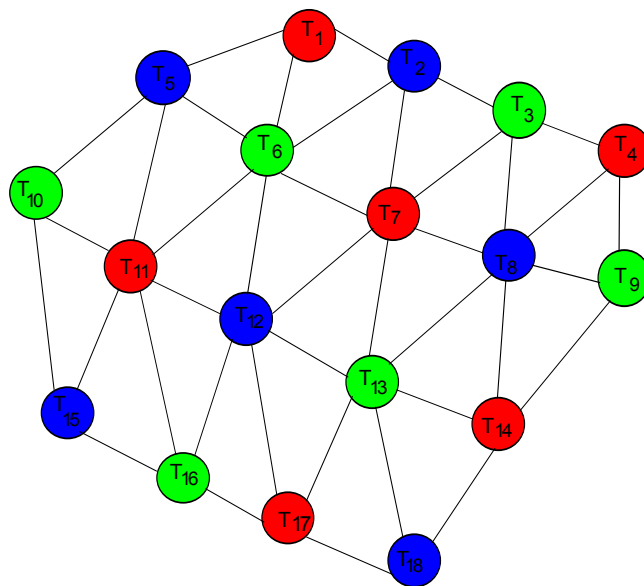
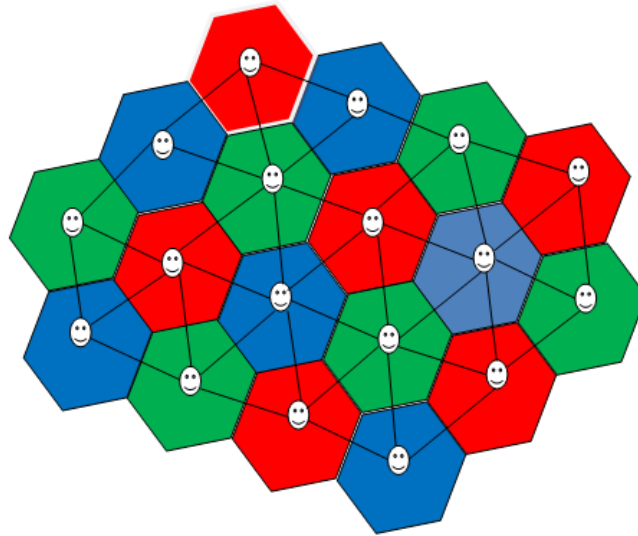


FIGURE 4.16 – Le graphe associé à Itération 3

comme tous les sommets sont colorés donc on arrête .

Après déroulement on a obtenu une coloration des cellules avec trois couleurs



A

Telque :

- La couleur une (1) est la couleur verte tel que $T(1)=\{6, 13, 3, 10, 9, 16\}$
- La couleur deux (2) est la couleur rouge tel que $T(2)=\{7, 1, 4, 11, 14, 17\}$
- La couleur trois (3) est la couleur bleu tel que $T(3)=\{8, 12, 2, 5, 15, 18\}$

La matrice d'adjacente associée au graphe présidant est :

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Après exécution du programme de Welsh and Powell en CodeBlocks sur cet exemple. Voici maintenant les résultats de son implémentation :

```

"C:\Users\SAMAH LINDA\Desktop\test1\main.exe"
entrez le nombre de cellule :
18
M[1][1]= 0
M[1][2]= 1
M[1][3]= 0
M[1][4]= 0
M[1][5]= 1
M[1][6]= 1
M[1][7]= 0
M[1][8]= 0
M[1][9]= 0
M[1][10]= 0
M[1][11]= 0
M[1][12]= 0
M[1][13]= 0
M[1][14]= 0
M[1][15]= 0
M[1][16]= 0
M[1][17]= 0
M[1][18]= 0
M[2][1]= 1
M[2][2]= 0
M[2][3]= 1
M[2][4]= 0
M[2][5]= 0
M[2][6]= 1
M[2][7]= 1
M[2][8]= 0
M[2][9]= 0
M[2][10]= 0
M[2][11]= 0
M[2][12]= 0
M[2][13]= 0
M[2][14]= 0
M[2][15]= 0
M[2][16]= 0
M[2][17]= 0
M[2][18]= 0
M[3][1]= 0
M[3][2]= 1
M[3][3]= 0
M[3][4]= 1
M[3][5]= 0
M[3][6]= 0
M[3][7]= 1
M[3][8]= 1
M[3][9]= 0
M[3][10]= 0
M[3][11]= 0
M[3][12]= 0
M[3][13]= 0
M[3][14]= 0
M[3][15]= 0
M[3][16]= 0
M[3][17]= 0

```

```

"C:\Users\SAMAH LINDA\Desktop\test1\main.exe"
M[18][2]= 0
M[18][3]= 0
M[18][4]= 0
M[18][5]= 0
M[18][6]= 0
M[18][7]= 0
M[18][8]= 0
M[18][9]= 0
M[18][10]= 0
M[18][11]= 0
M[18][12]= 0
M[18][13]= 1
M[18][14]= 1
M[18][15]= 0
M[18][16]= 0
M[18][17]= 1
M[18][18]= 0
Couleur 1 pour les sommets : 6 3 9 10 13 16
Couleur 2 pour les sommets : 7 1 4 11 14 17
couleur 3 : Pour le reste des sommets
Appuyez sur une touche pour continuer...

```

Après application directe sur le graphe modélisant le réseaux cellulaire vu précédemment, les résultats théorique coïncide avec les résultats obtenus, le graphe coloré est présenté dans la Figure 4.16 est 3-coriabie

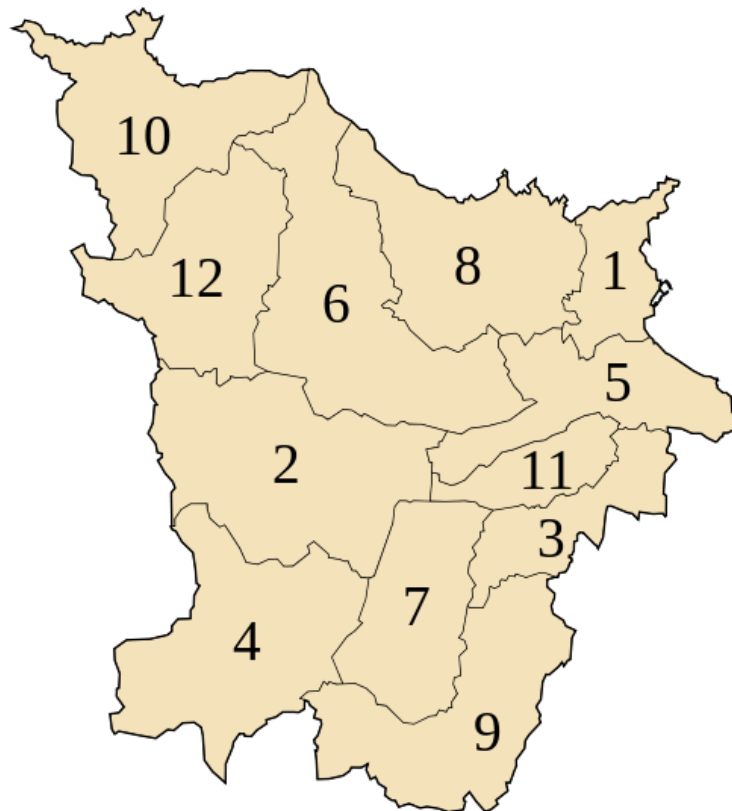
Ce programme marche pour la coloration de tous les graphes, ce n'est pas seulement pour les réseaux cellulaire , on peut colorer n'importe quelle carte, et organiser les examens de n'importe quelle établissement.

4.5 Application de la coloration de graphe pour la carte géographique des daïras de la wilaya de Annaba

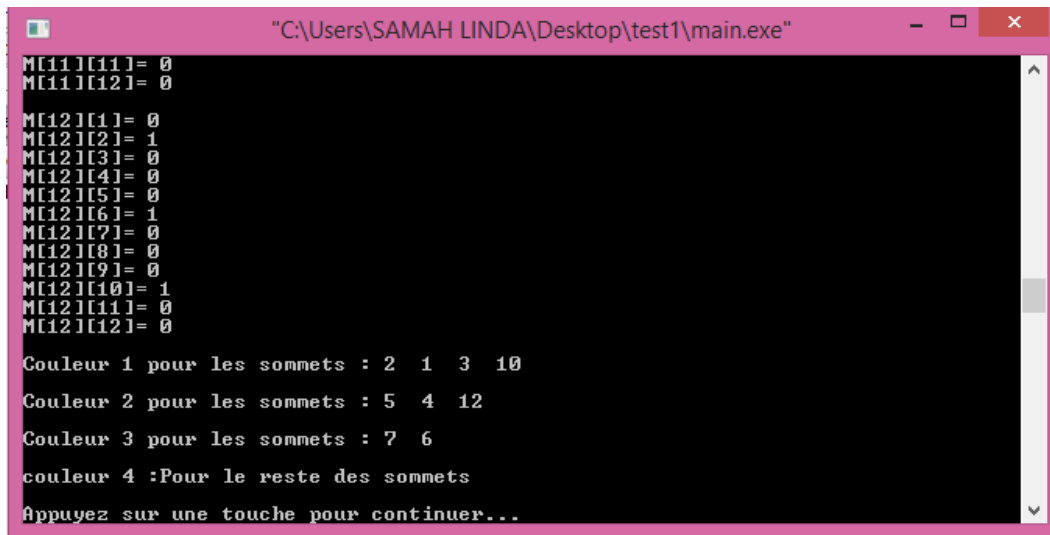
La wilaya de Annaba est composée de douze daïras (circonscriptions administratives)

Pour modéliser cette carte, les daïras vont être représenté par des sommets, et pour chaque deux daïras voisins, une arête relie ces deux daïras.

Chaque daïras possède un code respectif, à son tour, le code désigne le numéro du sommet ou le sommet qui revient sur la daïras.



Après exécution du programme de Welsh and Powell en CodeBlocks sur cet exemple, voici maintenant les résultats de son implémentation :



```
"C:\Users\SAMAH LINDA\Desktop\test1\main.exe"
M[1][1]= 0
M[1][2]= 0

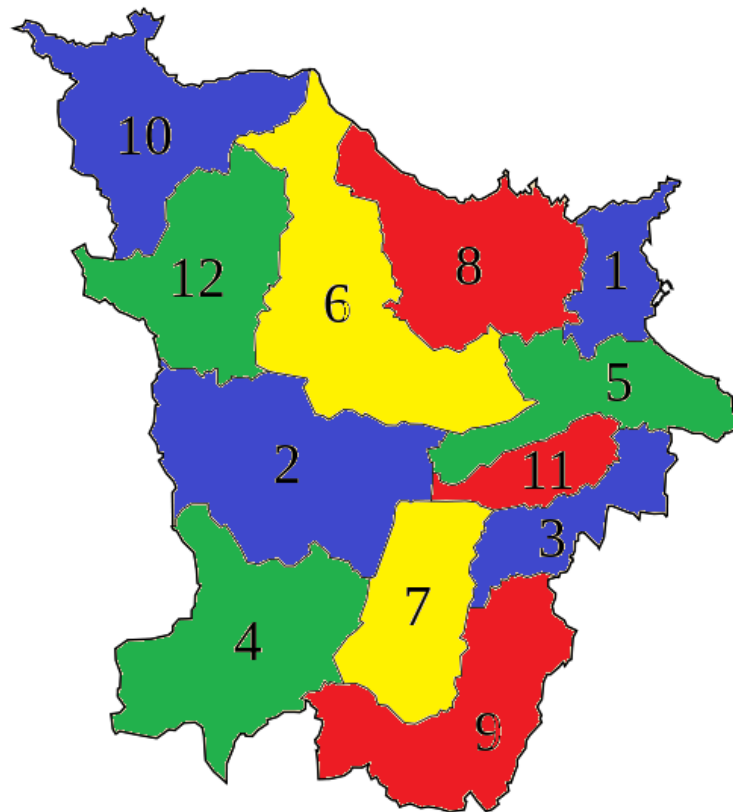
M[2][1]= 0
M[2][2]= 1
M[2][3]= 0
M[2][4]= 0
M[2][5]= 0
M[2][6]= 1
M[2][7]= 0
M[2][8]= 0
M[2][9]= 0
M[2][10]= 1
M[2][11]= 0
M[2][12]= 0

Couleur 1 pour les sommets : 2 1 3 10
Couleur 2 pour les sommets : 5 4 12
Couleur 3 pour les sommets : 7 6
couleur 4 :Pour le reste des sommets
Appuyez sur une touche pour continuer...
```

FIGURE 4.18 – Implémentation en CodeBlocks

Après application directe sur le graphe modélisant la carte géographique vu précédemment, voilà maintenant le graphe 4-coloré tel que :

- La couleur une (1) est la couleur bleu tel que $c(1)=\{2,1,3,10\}$
- La couleur deux (2) est la couleur vert tel que $c(2)=\{5,4,12\}$
- La couleur trios (3) est la couleur jaune tel que $c(3)=\{6,7\}$
- La couleur quatre (4) est la couleur rouge tel que $c(4) =\{8,9,11,12\}$



4.6 Conclusion

Nous avons consacré cette partie de notre travail, à une coloration par somme pour une classe de graphe et à un problème d'optimisation dans les graphes ceci avec application sur exemples. Le problème d'affectation de fréquences résolu de façon manuelle, et présentation d'une application de l'algorithme de Welsh and Powell sur un codeblocks que nous avons utilisé pour optimiser les interférences dans un réseau cellulaire et coloration d'une carte géographique.

Conclusion et perspectives

Ce travail présente à ses lecteur une vision générale de la théorie des graphes qui englobe un sujet très intéressant qui est la coloration dans les graphes.

L'importance de la théorie des graphe provient du fait qu'elle fournit un cadre conceptuel adéquat pour l'analyse et la résolution de nombreux problème.

Notre travail consiste à donné au lecteur un certain nombre d'outil (algorithmes) de la théorie des graphes directement utilisables pour résoudre des problèmes qui peuvent se poser à lui, et l'exactitude de notre objectif consiste à résoudre des problèmes en utilisant l'optimisation en théorie des graphes plus particulièrement la Coloration de graphes, pour cela on a opté pour la résolution d'un problèmes.

On a présenté au premier chapitre un rappel sur les graphes . Au deuxième chapitre on a étudié la Complexité algorithmique. Dans le troisième chapitre on a étudié la coloration des sommets et des arêtes d'un graphe et ces types, en utilisant les différent méthodes de résolution du problème dans le but de colorer les sommets du graphe.

Dans le quatrième chapitre on a traité quelques notions sur les réseaux cellulaire puis on a essayé de résoudre le problème d'allocation de fréquences en utilisant l'algorithme de Welsh-Powell. On a essayé dans ce projet de projeter les étapes de cet algorithme et la matrice d'adjacence du graphe, dans le but de colorer les sommets du graphe en utilisant le langage C.

Bibliographie

- [1] BELAÏD. A. *Algorithmique Avancée. Cours Master 1. Recherche Opérationnelle.* université de bejaia, 2015-2016.
- [2] M. Aigner and G. Ziegler. The australian mathematical society. *Proofs from the book*, page 127, 2003.
- [3] N. Bellharat. La théoré des graphe. Master's thesis, 2010.
- [4] C. Berge. *Graphes and hypergraphs.* Dunod Paris, 2nd edition, 1973.
- [5] A. J. Bondy and U. S. R. *Graphs Theory with Applications.* North Holland, 1976.
- [6] P. Bornsztien. Cours théorie des graphes, 2003.
- [7] T. Davot-Grangé. *Coloration de graphes par multi-sommes.* Mémoire mémoire master 2 en informatique, A Université de Bordeaux, 2018.
- [8] R. Diestel. *Graph theory, Graduate Texts in Mathematics*, volume 173. 2000.
- [9] R. Diestel. *Graph Thery.* springer, 2006.
- [10] Drira.K. Coloration d'êrtes ℓ -distance et clustering : etudes et algorithmes auto-stabilisants, École doctorale informatique et mathématiques. Master's thesis, université de lyon, 2010.
- [11] C. E. Berna fabrice. Peclier Thomas. *Algorithme distribué.*
- [12] Benjamin. K. *Modélisation, Optimisation, Complexité et algorithme.* 2005-2006.
- [13] W. Haken K. Appel and J. Koch. Every planar map is four colorable. part ii. reducibility. *Illinois J. Math.*, 21(3) :491–567.
- [14] K.Appel and W. Haken. Every planar map is four colorable. part i. discharging. *Illinois J. Math*, 21(3) :429–490.
- [15] J. Garvin Lonlac Konlac. Contributions 'a la r'ésolution du probl'eme de la satisfiabilité propositionnelle. Master's thesis, Doctorat de l'Universit 'e d'Artois, 03 juillet 2014.
- [16] B. KRAFT. Modélisation, optimisation, complexité et algorithmes. les graphes, leurs usages et leurs algorithmes. (1) :45–58, 2005.
- [17] F. Kramer. Sur le nombre chromatique $k(p,g)$ des graphes, rairo. (R-1) :67–70, (1972).

-
- [18] F. Kramer and H. Kramer. Un probleme de coloration des sommets d'un graphe. *C.R. Acad. Sci.*, pages 46–48, (1969).
- [19] K. Dalal et B. Reed L. Addario-Berry, R. Aldred. Vertex colouring edge partitions. *Journal of Combinatorial Theory, Series B*, page 237–244, 94 (2005).
- [20] T. Luczak et A. Thomason M. Karonski. Edge weights and vertex colours. *Journal of Combinatorial Theory, Series B*, page 151–157, 91 (2004).
- [21] Didier. Müller. *Introduction à la théorie des graphes*. CRM, 2012.
- [22] U. Montanari. Networks of constraints : Fundamental properties and applications to picture processing. *Artificial Intelligence*, 7 :55, 1974.
- [23] Grégory Morel. Stabilité et coloration des graphes sans p_5 . Thèse de doctorat, université de GRENOBEL.
- [24] J. Przybylo et M. Wozniak O. Baudon, J. Bensmail. On decomposing regular graphs into locally irregular subgraphs. *European Journal of Combinatorics*, page 90–104, 49(2015).
- [25] Eric Sopena. *Hommorphisme et Coloration de graphe*. 2007.

RÉSUMÉ

L'objectif de ce travail est de montrer l'utilité de la "théorie des graphes" pour l'optimisation dans les réseaux, et cela en prenant comme champs problème d'allocation de fréquences et la carte géographique. Ce mémoire contribue également à montrer un résultat théorique d'une nouvelle classe obtenue avec la coloration par somme des arêtes et particulièrement la coloration des sommets, pour un nombre chromatique minimal qui sera optimal. Pour cela une application de problème d'allocation de fréquences et la carte géographique est proposée, suivie d'une résolution, en faisant appel un programme réalisé sous CodeBlocks.

Mots-clés : Coloration de sommets ; Réseaux,.

ABSTRACT

The objective of this work is to show the usefulness of the "graph theory" for the optimization in networks and that can be seen as fields of frequency allocation problem and geographical map. This memory also contributes to show a theoretical result a new class obtained with sum coloring on edges and particularly the coloring of the vertices, of a minimal chromatic number that will be optimal. For this a frequency allocation problem application and map is proposed, following a resolution by doing call a program made under CodeBoks.

Keywords : Coloring of vertices ; Network .