

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

Université Akli Mohand Oulhadj - Bouira -



Faculté des Sciences et des Sciences Appliquées

Département de Mathématique

Mémoire de Master

Filière : Mathématiques

Spécialité : Recherche Opérationnelle

Thème

Problème du couplage dans un graphe

Présenté par :

Abchiche Chafiaa

Devant le jury d'examen composé de :

<i>M^{me}</i> Bekri Houria	Présidente	MAA	UAMO	Bouira
<i>M^{me}</i> Imine Ouiza	Promotrice	MAA	UAMO	Bouira
<i>M^{me}</i> Boudane Khadidja	Examinatrice	MAA	UAMO	Bouira
<i>M^{me}</i> Messaoudene karima	Examinatrice	MAA	UAMO	Bouira

2023/2024

Remerciements

Je tiens tout d'abord à remercier Dieu le Tout-Puissant, qui m'a donné la force et la patience d'accomplir ce travail dans les meilleures conditions.

Je tiens à exprimer toute ma reconnaissance à Madame Imine Ouiza. Je la remercie de m'avoir encadré, orienté, aidé et conseillé pendant la réalisation de ce travail.

J'adresse mes sincères remerciements à tous les enseignants du département de mathématiques et à toutes les personnes qui, par leurs conseils et leurs critiques, ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.

Je souhaite également remercier les membres du jury pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critiques. À toutes les personnes intervenantes, de loin ou de près, pour l'élaboration de ce travail, je présente mes remerciements, mon respect et ma gratitude.

Enfin, je remercie mes parents et mes proches pour leur soutien moral.

Dédicace

Je dédie ce modeste travail à :

Mes très chers parents, qui n'ont jamais cessé
de m'encourager et me soutenir,
et mes chères sœurs et frères.

Toute ma famille sans exception,
et mes amis.

Table des matières

- Introduction Générale** **9**

- 1 Notion des bases** **11**
 - 1.1 Introduction : 11
 - 1.2 Définitions et notations : 11
 - 1.3 Les graphes particuliers : 18
 - 1.3.1 Graphes complet : 18
 - 1.3.2 Graphes complémentaire : 18
 - 1.3.3 Graphe biparti : 19
 - 1.4 Conclusion : 26

- 2 Résolution du problème de couplage** **27**
 - 2.1 Introduction 27
 - 2.2 Définition et premières propriétés 27
 - 2.2.1 Chemin augmentant/alternant [1] 28
 - 2.3 Les types du couplage 29
 - 2.3.1 Couplage maximal 29
 - 2.3.2 Couplage maximum 29
 - 2.3.3 Couplage parfait 37
 - 2.3.4 Couplage dans graphe biparti 38
 - 2.3.5 Couplage d'un graphe non biparti 44
 - 2.3.6 Couplage d'un graphe orienté 45
 - 2.3.7 Couplage d'un graphe non orienté 45

2.3.8	Couplage pondéré	46
2.4	Les propriétés du couplage	46
2.5	Les intérêts du couplage	46
2.6	Conclusion	47
3	Complexité des algorithmes	48
3.1	Introduction	48
3.2	Analyse de complexité temporelle des algorithmes	48
3.3	Comparaison de performances et des caractéristiques des algorithmes . . .	50
	Conclusion générale	51
	Résumé	54
	Abstract	55

Liste des tableaux

2.1	La différence entre le couplage maximum et maximal	32
-----	--	----

Table des figures

Fig-1	12
Fig-2	12
Fig-3	13
Fig-4	14
Fig-5	14
Fig-6	15
Fig-7	16
Fig-8	16
Fig-9	17
Fig-10	17
Fig-11	18
Fig-12	18
Fig-13	19
Fig-14	19
Fig-15	20
Fig-16	20
Fig-17	21
Fig -18	22
Fig -19	23
Fig -20	24
Fig -21	24
Fig -22	25
Fig-23	25

Fig -24	28
Fig-25	28
Fig-26	28
Fig-27	29
Fig-28	29
Fig-29	30
Fig-30	31
Fig-31	33
Fig-32	33
Fig-33	34
Fig-34	34
Fig-35	34
Fig-36	35
Fig-37	36
Fig-38	36
Fig-39	38
Fig-40	38
Fig-41	39
Fig-42	44
Fig-43	45
Fig-44	45

Introduction Générale

La recherche opérationnelle est un domaine qui vise à développer et appliquer des méthodes analytiques avancées pour résoudre des problèmes complexes d'optimisation et d'aide à la décision. Elle fait appel à diverses techniques mathématiques, statistiques et algorithmiques pour modéliser et optimiser des systèmes réels.

L'un des outils fondamentaux de la recherche opérationnelle est la théorie des graphes, qui étudie les propriétés des graphes, ces structures mathématiques composées de sommets reliés par des arêtes. Les graphes permettent de représenter et d'analyser de nombreux types de systèmes et de problèmes concrets.

Un problème classique et important en théorie des graphes est le problème du couplage. Ce dernier est un ensemble d'arêtes sans arêtes adjacentes. Ce problème consiste à trouver un couplage de cardinalité maximale, c'est-à-dire contenant le plus grand nombre possible d'arêtes.

De manière générale, le problème du couplage intervient dans de nombreuses applications pratiques nécessitant d'apparier ou d'assigner des ressources de façon optimale. On le retrouve notamment en logistique, affectation de tâches, théorie des jeux, chimie organique, télécommunications ou encore en bio-informatique.

La difficulté du problème réside dans le fait qu'il faut évaluer toutes les combinaisons d'arêtes possibles pour identifier le couplage optimal. Plusieurs algorithmes ont été développés, avec des complexités variables selon la structure du graphe. Dans le cas général, c'est un problème NP-difficile.

L'étude des couplages reste un sujet de recherche actif, que ce soit pour améliorer les algorithmes existants, développer de nouvelles techniques de résolution approchées ou traiter des variantes avec contraintes supplémentaires.

Au premier chapitre, nous présentons les définitions et la terminologie adoptée en théorie des graphes en général.

Dans le deuxième chapitre nous introduisons en détail les différents aspects de ce problème et présenter les principales méthodes de résolution algorithmiques.

L'objet de troisième chapitre est d'analyser en détail la complexité temporelle des principaux algorithmes pour résoudre le problème du couplage dans les graphes. Nous comparerons leurs performances théoriques et leurs caractéristiques, en mettant en évidence leurs forces et leurs faiblesses respectives. Cette analyse est essentielle pour choisir l'algorithme le plus approprié en fonction de la taille et de la structure de l'instance du problème à résoudre.

Enfin nous terminons notre travail par une conclusion générale sur les différentes méthodes utilisées en problème du couplage.

Chapitre 1

Notion des bases

1.1 Introduction :

Dans ce chapitre, nous présentons les différents concepts et notions, particulièrement destinés aux lecteurs qui ne sont pas familiers avec la théorie des graphes en général. Ces définitions sont fournies pour aider le lecteur à mieux comprendre le contenu de ce travail. Comme les chaînes les chemins les cycles les circuits ...etc.

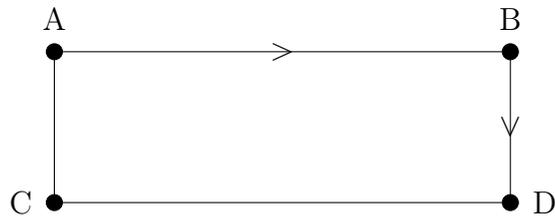
Ce chapitre offre une base solide pour saisir les structures et les propriétés des graphes.

1.2 Définitions et notations :

Définition 1.1. *un graphe*

Un graphe est un dessin géométrique défini par la donnée d'un ensemble de points (appelés sommets) reliés entre eux par des lignes ou flèches (appelées arêtes ou arcs respectivement)

Exemple 1.



– Fig-1

(A,B,C,D) sont des sommets

(A,C) est une arête

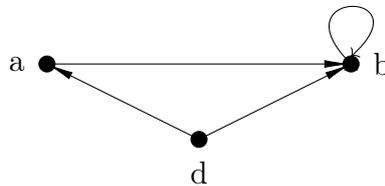
(A,B) est un arc

Définition 1.2. graphe orienté

Un graphe orienté est un graphe défini par un ensemble des sommets $X = (x_1, x_2, \dots, x_n)$ reliés par des arcs seulement .

- On note $E = (e_1, e_2, \dots, e_m)$ l'ensemble des arcs $|E| = m$
- On note un graphe orienté par $G = (X, E)$
- On note un arc reliant un sommet x_1 au sommet x_2 par $e = (x_1, x_2)$ avec x_1 est l'extrémité initiale $I(e)$ et x_2 l'extrémité terminale notée $T(e)$.

Exemple 2.



– Fig-2

$e(d,a)$ est un arc avec $I(e) = d$ et $T(e) = a$

Remarque 1.2.1.

-Dans un graphe orienté $(x_1, x_2) \neq (x_2, x_1)$

-l'arc $e = (x_1, x_1)$ est appelé une boucle.

Définition 1.3. l'ordre d'un graphe

Soit $G = (X, E)$, si $|X| = n$, on dit que le graphe G est d'ordre n .

Dans le graphe précédant l'ordre de G est $n=3$

Définition 1.4. graphe non orienté

Un graphe non orienté est un graphe défini par l'ensemble des sommets $X = \{x_1, x_2, \dots, x_n\}$

reliés par des arêtes seulement on le note $G = (X, E)$ ou $X = \{x_1, x_2, \dots, x_n\}$ $E =$

$\{e_1, e_2, \dots, e_m\}$ $|X| = n, |E| = m$

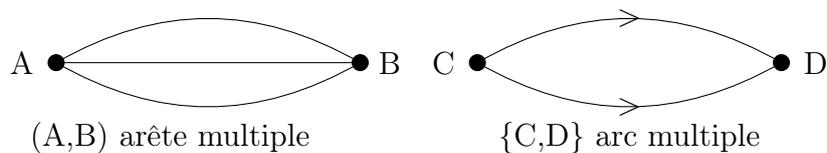
Remarque 1.2.2. Dans un graphe non orienté $(x_1, x_2) = (x_2, x_1)$ l'ordre n'est pas important.

Définition 1.5. graphe simple et graphe multiple

— Un graphe simple est un graphe sans boucle ni arête (arc) multiple.

— Un graphe multiple est un graphe qui possède des boucles et des arêtes (arcs) multiples

Exemple 3.



– Fig-3

Degré d'un sommet :

Soit $G = (X, E)$ un graphe orienté

— Le demi-degré extérieur d'un sommet x égal au nombre d'arcs ayant le sommet x comme extrémité initial .

on le note $d_g^+(x) = |\{e \in E / I(e) = x\}|$

— Le demi degré intérieur d'un sommet x est égal au nombre d'arcs ayant x comme extrémité terminale .

on le note $d_g^-(x) = |\{e \in E / T(e) = x\}|$

— Le degré d'un sommet x est égal au nombre d'arcs ayant x comme extrémité initiale ou terminale

On dit aussi que le degré de x est le nombre d'arcs adjacents a x , on le note $d_G(x) = d_G^+(x) + d_G^-(x)$.

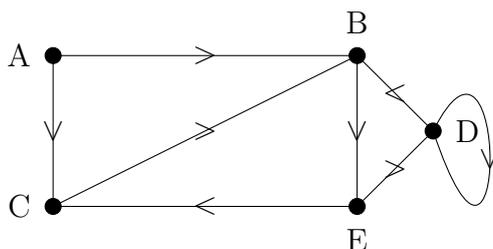
Remarque 1.2.3. *une boucle apporte une contribution de deux dans le calcul du degré d'un sommet.*



– Fig-4

le degré de A est : $d^+(A) + d^-(A) = 1 + 1 = 2$

Exemple 4.



– Fig-5

$$d_G^+(A) = 2 \quad \rightarrow d_G(A) = 2$$

$$d_G^-(A) = 0$$

$$d_G^+(B) = 1 \quad \rightarrow d_G(B) = 4$$

$$d_G^-(B) = 3$$

$$\begin{aligned} d_G^+(D) = 2 & \rightarrow d_G(D) = 4 \\ d_G^-(D) = 2 & \end{aligned}$$

Propriété 1.2.1. Soit un graphe orienté $G = (X, E)$

la somme de demi-degré intérieurs des sommets de G est égale à la somme des demi-degrés extérieurs des sommets de G .

$$\sum_{x \in X} d_G^+(x) = \sum_{x \in X} d_G^-(x)$$

Dans chaque graphe la somme des degré est un nombre pair et ona .

$$\sum_{x \in X} d(x) = 2m$$

m : est le nombre d'arcs ou arêtes.

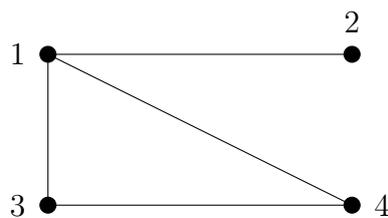
Notation 1.1.

On note le plus grand degré des sommets d'un graphe par $\Delta(G)$ et le plus petit degré des sommets par $\delta(G)$.

$$\Delta(G) = \max_{x \in G} d(x)$$

$$\delta(G) = \min_{x \in G} d(x).$$

Exemple 5.



– Fig-6

$$\Delta(G) = \max(3, 1, 2, 2) = 3 = d(1).$$

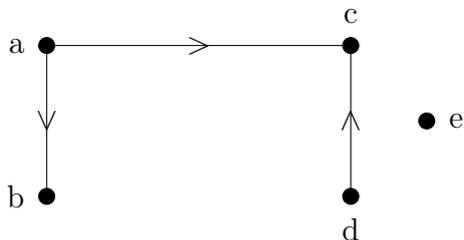
$$\delta(G) = \min(3, 1, 2, 2) = 1 = d(2).$$

Définition 1.6. sommet pendant sommets isolé :

Un sommet pendant est un sommet dont le degré égal 1 ($d_G(x) = 1$) .

Un sommet isolé est un sommet dont le degré égal a zéro ($d_G(x) = 0$) .

Exemple 6.



– Fig-7

Les sommets b et d sont des sommets pendants $d_G(b) = d_G(d) = 1$.

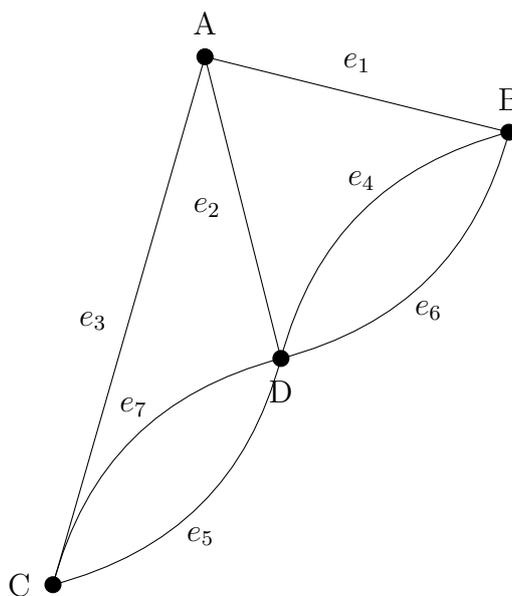
Le sommet e est isole, $d(e) = 0$.

Définition 1.7. sous graphe, graphe partiel

Soit $G = (X, E)$ un graphe

1. soit $A \subset X$, on appelle sous graphes de G engendré par A le graphe $G_A = (A, E_A)$.
2. soit $W \subset E$, on appelle sous graphes de G engendré par W le graphe $G_W = (X, W)$.

Exemple 7.

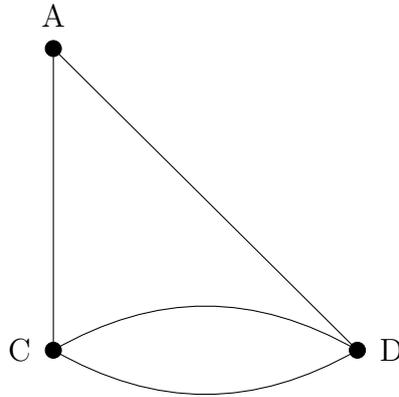


– Fig-8

$$A = \{A, D, C\}$$

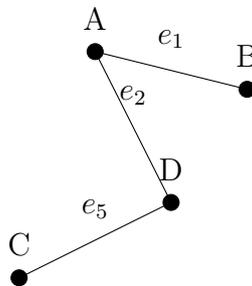
$$W = \{e_1, e_2, e_5\}$$

Le sous graphe engendré par A est :



– Fig-9

Le graphe partiel engendré par W est :

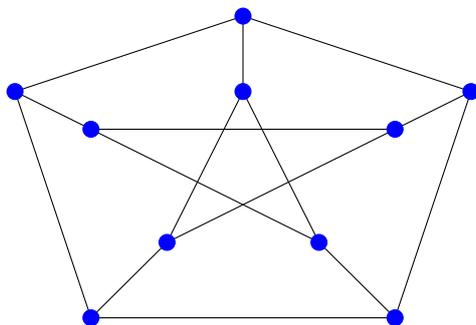


– Fig-10

Définition 1.8. *graphe cubique* :

Un graphe cubique est un type de graphe régulier caractérisé par un degré de 3 pour chaque sommet. Autrement dit, dans ce type de graphe, chaque sommet est relié exactement à trois arêtes distinctes.

Exemple 8.



– Fig-11

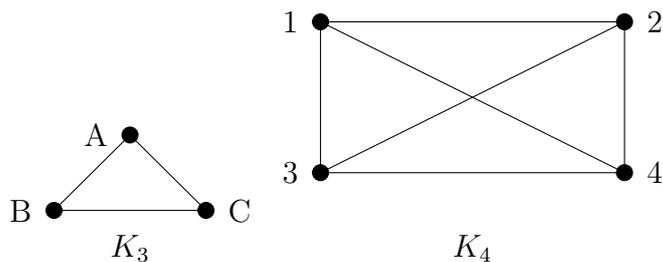
Ce graphe est un graphe cubique car tous les sommets sont de degré (3)

1.3 Les graphes particuliers :

1.3.1 Graphes complet :

Un graphe complet est un graphe dans lequel chaque paire de sommets est reliée par une arête on le note K_n .

Exemple 9.

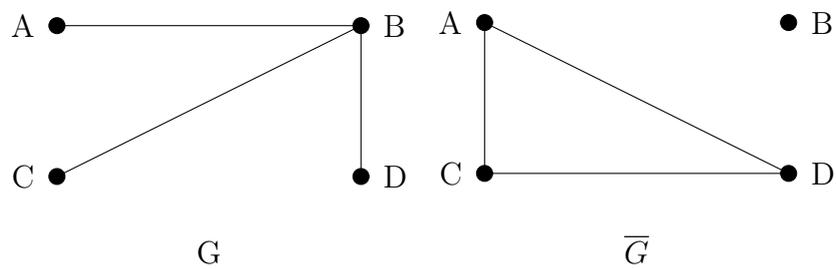


– Fig-12

1.3.2 Graphes complémentaire :

Le graphe complémentaire d'un graphe simple $G = (X, E)$ est défini comme suit :
 $u \in E \Leftrightarrow u \notin \bar{E}$. (noté $\bar{G} = (X, \bar{E})$). En d'autres termes, les arêtes qui ne sont pas présentes dans G sont présentes dans son graphe complémentaire, et vice versa.

Exemple 10.



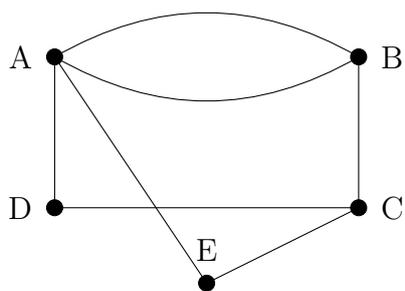
– Fig-13

Remarque 1.3.1. $\bar{G} \cup G$ est un graphe complet.

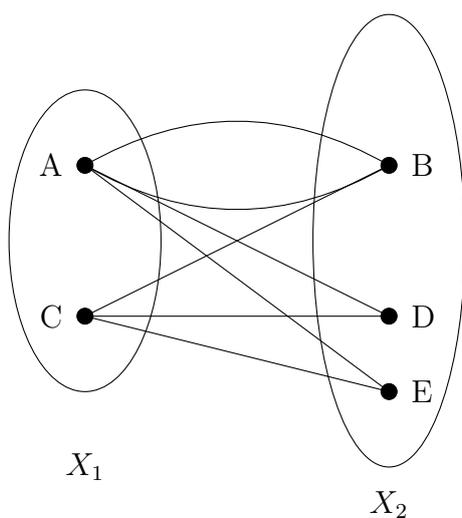
1.3.3 Graphe biparti :

Un graphe est dit biparti si l'ensemble de ses sommets peut être réparti en deux ensembles X_1, X_2 de sorte que deux sommets appartenant au même ensemble ne soient pas adjacents. On le note souvent par $G = (X_1, X_2, E)$

Exemple 11.



– Fig-14



– Fig-15

$$X_1 = \{A, C\}$$

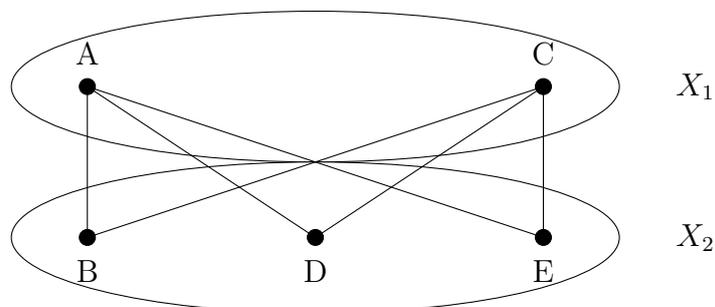
$$X_2 = \{B, D, E\}$$

$G = (X_1 \cup X_2, E)$ est un graphe biparti.

Remarque 1.3.2.

- Un graphe biparti est complet si chaque sommet de X_1 est adjacents a tout sommet de X_2
- Un graphe simple biparti complet est note $K_{p,q}$
 $p = |X_1|$.
 $q = |X_2|$.

Exemple 12.



– Fig-16

G est un graphe biparti complet $K_{2,3}$.

Algorithme de K - coloration d'un graphe :

On appelle K coloration d'un graphe $G = (X, E)$ une partition de l'ensemble des sommets X de G en k -classe X_1, X_2, \dots, X_k de telle façon que deux sommets d'une même classe sont colories de la même couleur .

Algorithme :

On commence par établir une liste ordonnée des sommets (ordonnée les sommets suivant l'ordre décroissant de leur degré).

1. Choisir une couleur appelle couleur d'usage.
2. Chercher dans la liste des sommets le premier sommet non cherché et le colorer avec la couleur d'usage.
3. Colories chaque sommet non adjacent à un sommet déjà colore avec la couleur d'usage
4. S'il reste des sommets non colore aller a 1

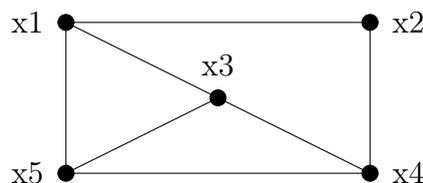
Définition 1.9.

On appelle nombre chromatique d'un graphe le plus petit nombre K pour lequel le graphe G est K -coloriable. On le note $\gamma(G)$.

Remarque 1.3.3.

Dans un graphe biparti $\gamma(G) = 2$.

Exemple 13.

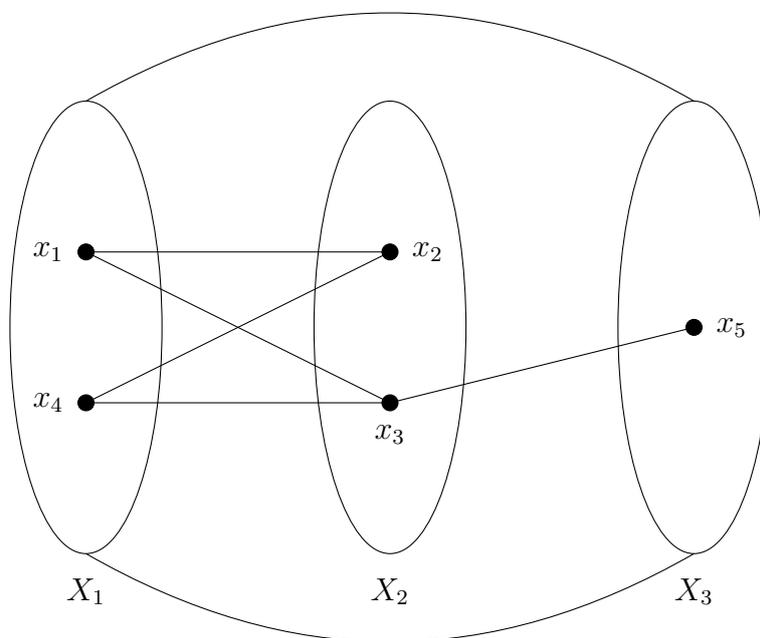


- Fig-17

La liste ordonnée des sommets $x_1 - x_3 - x_4 - x_5 - x_2$

1. Colorie le sommet x_1 par la couleur verte, puis cherchons dans la liste, le sommet non coloré qui n'est pas adjacents à aucun sommet coloré et le coloré avec la couleur verte, soit le sommet x_4 la liste ordonnée devient : $x_3 - x_5 - x_2$
2. Colorier le sommet x_3 par la couleur rouge puis le sommet x_2 . La liste ordonnée devient x_5 .
3. Il reste le sommet x_5 colorons le avec la couleur bleu, donc le graphe est 3 coloriable et on peut le partitionné en 3 classes :

$$X_1 = \{x_1, x_4\}, X_2 = \{x_2, x_3\}, X_3 = \{x_5\}. \text{ donc } \gamma(G) = 3.$$



– Fig -18

Définition 1.10. la chaîne :

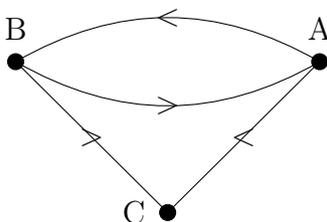
La chaîne reliant deux sommets x_0 et x_k dans un graphe G est une séquence des sommets reliés par des arêtes ,telle que deux sommets successifs partagent une arête commune. On la note (x_0, x_1, \dots, x_k) on dit que x_0 et x_k sont les extrémités de la chaîne .

Remarque 1.3.4.

— Une chaîne est dite simple si on passe une fois par ses arêtes.

— Dans une chaîne l'orientation n'est pas importante.

Exemple 14.



– Fig -19

— (A, B, C) est une chaîne simple

— (A, C, B, A, C) n'est pas simple car l'arc (A, C) est parcouru deux fois .

Définition 1.11. Chemin :

Soit $G = (X, E)$ un graphe orienté.

Un chemin est une suite de sommets reliés successivement par des arcs orientés dans le même sens ou le note : (x_0, x_1, \dots, x_k) .

Exemple 15. (dans exemple précédente :)

— (A, B, C) est un chemin simple .

— (A, B, A, B, C) est un chemin pas simple .

Car l'arc AB est parcouru deux fois .

Définition 1.12. Cycle :

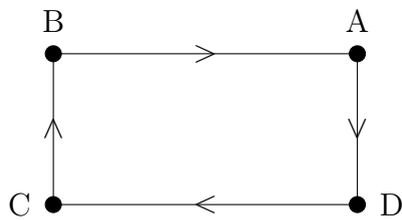
Un cycle est une chaîne simple dont les extrémités coïncident on le note $(x_0, x_1, \dots, x_k = x_0)$.

Définition 1.13. Circuits :

Circuit est un chemin dont les extrémités sont confondues, on le note : $(x_0, x_1, \dots, x_k = x_0)$.

Remarque 1.3.5. Une boucle est un cycle (circuit) particulier .

Exemple 16.



– Fig -20

- (A,B,C,D,A) est un cycle.
- (B,A,D,C,B) est un circuit .

Définition 1.14. arbre :

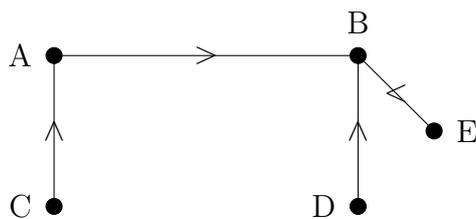
Soit $G = (X, E)$ un graphe tel que $|X| = n, |E| = m$.

G est un arbre si G est connexe et sans cycle par conséquence : $m = n - 1$

Car : G est connexe $\Rightarrow m \geq n - 1$

G est sans cycle $\Rightarrow m \leq n - 1$

Exemple 17.



– Fig -21

Le graphe G est un arbre car connexe et sans cycle.

Remarque 1.3.6. Un arbre est un graphe simple ayant $n-1$ arcs.

Transversal :

Soit $G = (X, E)$ un graphe .

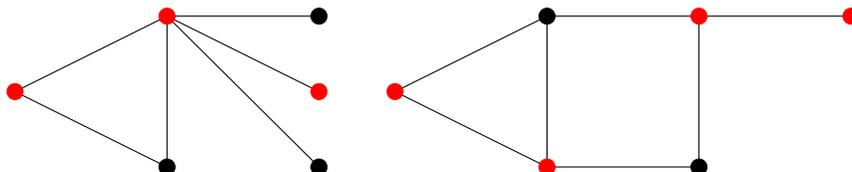
$T \subset E$ est un transversal (ou couverture de sommets) de G si toute arête de X possède,

au moins, une extrémité dans T

Un transversal T de G est minimal si $T - \{e\}$, $\forall e \in T$ n'est pas un transversal de G .

Un transversal T de G est minimum si $|T| \leq |T'|, \forall T'$ transversal de G .

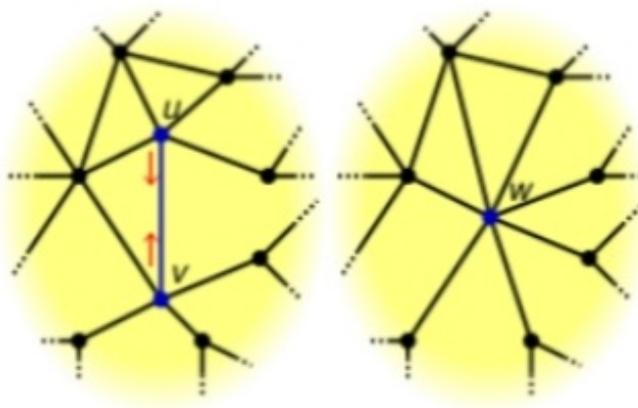
Exemple 18.



- Fig -22

Graphe contracté :

La contraction d'une arête dans un graphe simple non orienté $G = (X, E)$ consiste à fusionner les deux sommets extrémités de l'arête $e = (u, v)$.



- Fig-23

La contraction d'un chemin consiste à réaliser des contractions successives de toutes les arêtes de ce chemin. Le graphe obtenu après ces contractions est le graphe résultant.

1.4 Conclusion :

Cette étude des graphes a permis de comprendre la nature complexe de ces entités mathématiques et leur importance fondamentale dans la modélisation et la résolution de problèmes pratiques.

Chapitre 2

Résolution du problème de couplage

2.1 Introduction

Dans ce chapitre on s'intéresse à la notion du couplage dans les graphes, en mettant en lumière les définitions, les chemins augmentants, les propriétés, et les différents types de couplages. On explore également l'algorithme d'Edmonds pour trouver des couplages maximaux, ainsi que l'algorithme hongrois pour résoudre les problèmes d'affectation minimale ou maximale dans les graphes bipartis. Des exemples concrets et des explications détaillées sont fournis pour illustrer ces concepts et méthodes algorithmiques. Ce chapitre couvre également des aspects avancés tels que les couplages pondérés, les couplages dans les graphes orientés et non orientés, ainsi que les propriétés et les applications des couplages dans divers domaines.

2.2 Définition et premières propriétés

Définition 2.1. *Dans un graphe simple non orienté $(G = (X, E))$, un couplage est un ensemble d'arêtes tel que chaque arête n'a pas de sommets en commun avec les autres arêtes du couplage.*

- En d'autres termes, les arêtes d'un couplage ne partagent pas de sommets communs.

2.2.1 Chemin augmentant/alternant [1]

- Un chemin alternant est un chemin qui alterne entre des arêtes appartenant à un couplage et des arêtes appartenant au reste du graphe (E/C).



– Fig -24

- • Un chemin augmentant est un chemin alternant qui commence et se termine par des sommets non appariés. Si un chemin augmentant est trouvé, cela signifie que le couplage n'est pas maximum.

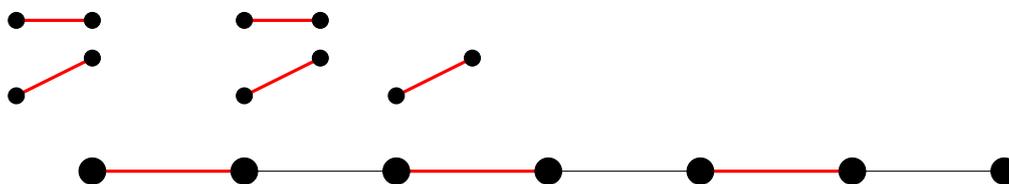


– Fig-25

Remarque 2.2.1.

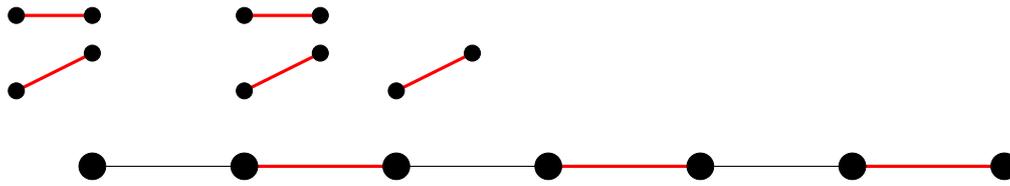
Les sommets non appariés sont des sommets qui ne font pas partie d'un couplage dans un graphe, c'est-à-dire qu'ils ne sont connectés à aucun autre sommet par une arête du couplage.

Propriété 2.2.1. [1] *Un couplage C est une collection d'arêtes en rouge qui contient un chemin augmentant P .*



– Fig-26

Un nouveau couplage $C' = C \Delta P$ qui possède plus d'arêtes que C .



– Fig-27

Remarque 2.2.2. *Si un chemin augmentant existe pour le couplage C , cela signifie que nous pouvons augmenter le nombre d'arêtes dans le couplage en utilisant ce chemin. Par conséquent, le couplage C n'est pas maximum.*

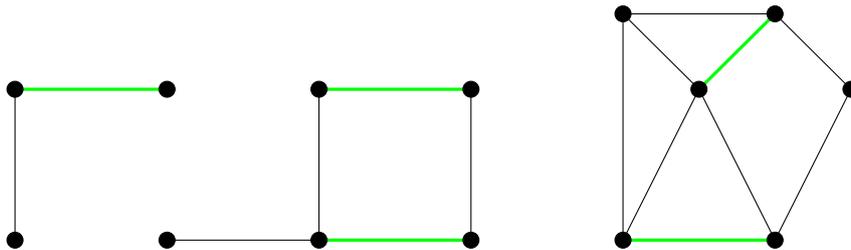
2.3 Les types du couplage

En théorie des graphes, il existe plusieurs types de couplages, tels que :

2.3.1 Couplage maximal

Un couplage maximal est un couplage qui contient le plus grand nombre possible d'arêtes. Un graphe peut avoir plusieurs couplages maximaux.

Exemple 19.



– Fig-28

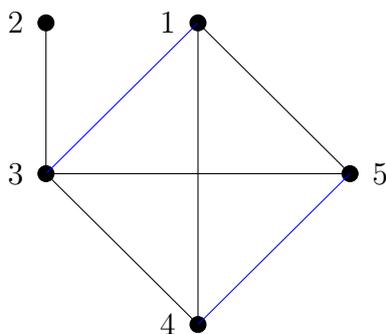
— En vert, un couplage maximale est de cardinalité $|C| = 5$

2.3.2 Couplage maximum

Est un ensemble d'arêtes dans un graphe qui contient le plus grand nombre possible d'arêtes, c'est-à-dire qu'il a une cardinalité maximale ($|C|$). En d'autres termes, un cou-

plage maximum est celui qui couvre autant de sommets que possible sans partager de sommets communs entre les arêtes incluses.

Exemple 20.



– Fig-29

— En bleu , un couplage maximum est de cardinalité $|C| = 2$

Algorithme d'Edmonds

L'algorithme d'Edmonds, également connu sous le nom d'« algorithme des fleurs et des pétales », est utilisé pour construire des **couplages maximaux** dans les graphes. Voici comment il fonctionne :

1. Définition du problème :

- o Étant donné un graphe $G = (X, E)$, l'algorithme cherche un couplage C tel que chaque sommet de X soit incident à au plus une arête dans E , et C soit de cardinal maximal.
- o Contrairement au couplage biparti, cet algorithme se base sur l'idée qu'un cycle de longueur impaire dans le graphe (appelé fleur) peut être contracté en un seul sommet.

2. Chemin augmentant :

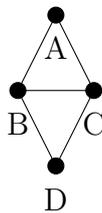
- o Un chemin alternant dans G passe alternativement par des arêtes de C et des arêtes de E/C .
- o Un chemin augmentant est un chemin alternant qui commence et se termine par deux sommets non couverts par C .

- o L'algorithme améliore itérativement un couplage initial vide le long de chemins augmentant dans le graphe.

3. Algorithme d'Edmonds :

- o À partir d'un couplage initial, l'algorithme calcule un couplage maximal en augmentant le couplage actuel avec des chemins augmentant tant que possible.
- o Si aucun chemin augmentant n'est trouvé, il renvoie le couplage obtenu.

Exemple 21.



– Fig-30

1. Initialisation :

- o Nous commençons avec un couplage vide (aucune arête sélectionnée) : $C = \{\emptyset\}$.
- o Le graphe initial est celui décrit ci-dessus.

2. Recherche de chemins augmentants :

- o Nous cherchons des chemins alternants qui commencent et se terminent par des sommets non couverts par C .
- o Supposons que nous trouvons le chemin augmentant $\{(B,C,D)\}$ (les sommets en gras ne sont pas encore couverts par C).

3. Augmentation du couplage :

- o Nous remplaçons l'arête $\{(B,C)\}$ par $\{(B,D)\}$ dans C .
- o Le nouveau couplage est donc $C = \{(B,D)\}$.

4. Répétition :

- o Nous continuons à chercher des chemins augmentants jusqu'à ce qu'il n'y en ait plus.

- o Supposons que nous trouvions un autre chemin augmentant $\{(A,B,C)\}$.
- o Nous remplaçons l'arête $\{(B,D)\}$ par $\{(A,B)\}$ dans C.
- o Le nouveau couplage est maintenant $C = \{(A,B), (C,D)\}$.

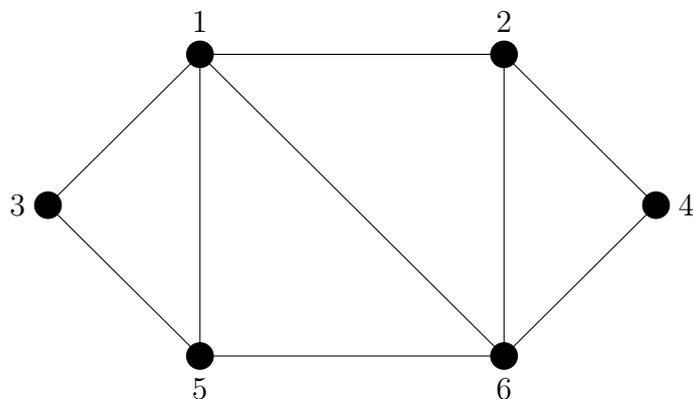
La différence entre le couplage maximum et maximal :

Le couplage maximal	Le couplage maximum
Un couplage (C) est considéré comme maximal lorsqu'aucune arête supplémentaire ne peut être ajoutée à cet ensemble sans violer sa propriété de couplage.	Qui contient le plus grand nombre possible d'arêtes
Il est possible qu'un graphe peut avoir plusieurs couplages maximaux.	C'est le couplage le plus grand dans le graphe, c'est-à-dire celui qui couvre le plus grand nombre de sommets possible sans que les arêtes incluses ne partagent de sommets communs.
	Contrairement au couplage maximal, un graphe donné ne possède qu'un seul couplage maximum.

TABLE 2.1 – La différence entre le couplage maximum et maximal

Remarque 2.3.1. *un couplage maximal est un ensemble d'arêtes qui ne peut pas être agrandi, tandis qu'un couplage maximum est celui qui comporte le plus grand nombre d'arêtes possible. Autrement dit, un couplage maximal est déjà aussi grand qu'il peut l'être, tandis qu'un couplage maximum est le plus grand parmi tous les couplages possibles dans le graphe.*

Exemple 22.



– Fig-31

Les arêtes (1,5) et (2,6) forment un couplage maximal.

Les arêtes (3,5), (1,6) et (2,4) forment un couplage maximum.

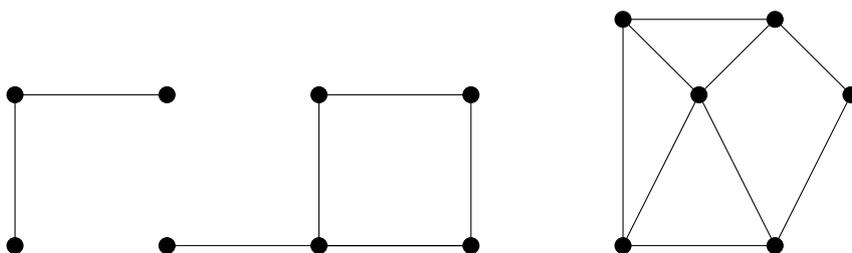
Chemin augmentant et couplage maximum :

Théorème 2.1. (Berge 1957) *Un couplage C dans un graphe G est considéré comme maximum si et seulement s'il n'existe pas de chemin augmentant.*

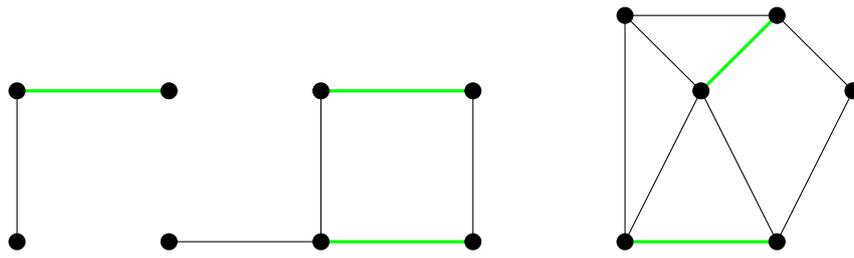
Preuve 2.1.

- Soit (C) et (C') deux couplages, où (C) est non-maximum et (C') est maximum.
- Soit H le sous-graphe de G défini par les arêtes de $C \Delta C'$.

Les composantes connexes de (H) sont soit des cycles, soit des chemins, et les arêtes de (C) et (C') alternent. Donc on a le graphe G de Fig-32 .

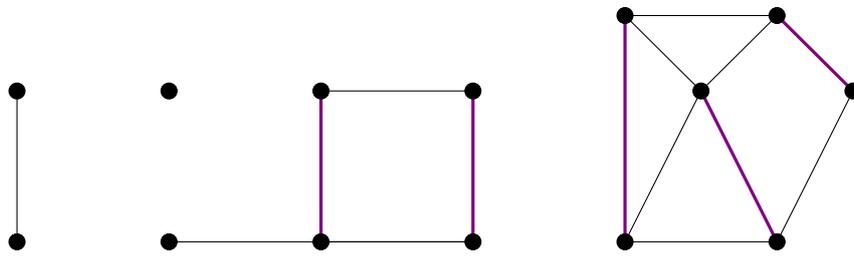


– Fig-32



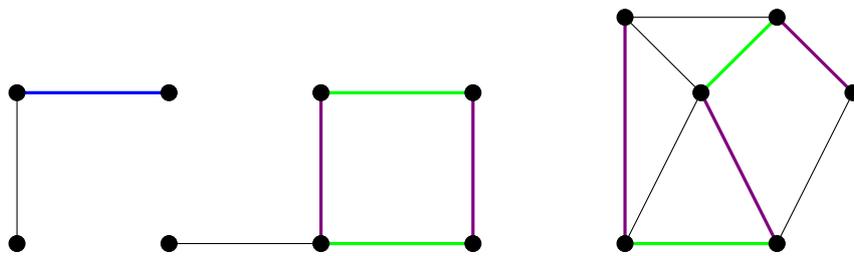
- Fig-33

$C =$ arêtes vertes



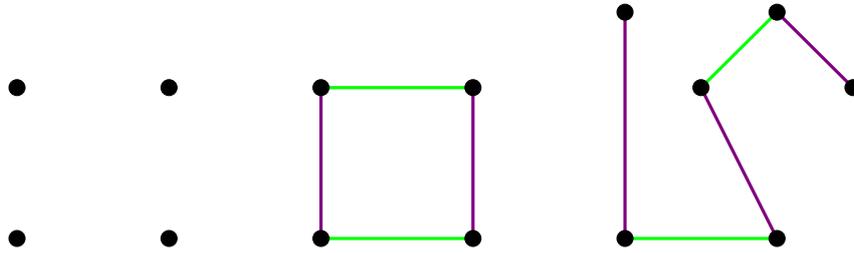
- Fig-34

$C' =$ arêtes violettes



- Fig-35

Le sous-graphe H



– Fig-36

Donc on a :

- Pour chaque cycle : si (C) est un couplage non-maximum et (C') est un couplage maximum, alors le nombre d'arêtes dans (C') est le même que celui dans (C) .
- Pour un chemin P : le nombre d'arêtes de C' supérieur au nombre d'arêtes de C . $(C'$ a plus d'arêtes que C). le chemin (P) commence et se termine par deux arêtes de (C) .

En effet, le chemin (P) est un chemin augmentant pour le couplage (C) .

Définition 2.2. Blossom(fleur)

Un cycle B est un fleur si :

- Il contient $2k + 1$ arêtes(où k est un entier).
- K de ses arêtes appartiennent au couplage C .

Couplage et contraction :

Théorème 2.2. [Edmonds 1965] : Soit C un couplage dans un graphe G .

Supposons que B est un blossom de G .

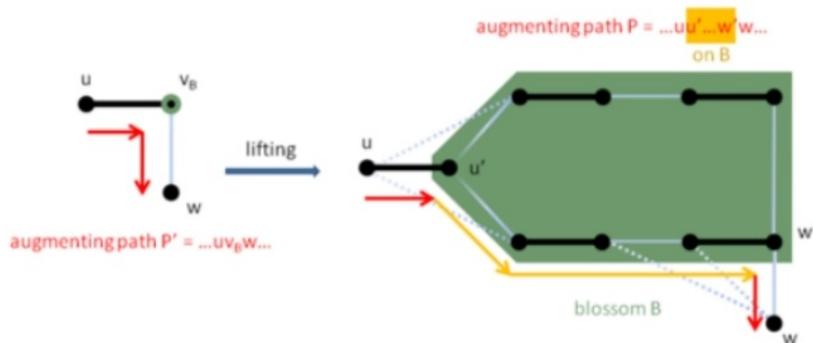
Soit G' le graphe obtenu par la contraction de B dans G .

Soit C' le couplage de C projeté dans G' .

G' a un chemin augmentant pour C' si et seulement si G a un chemin augmentant pour C .

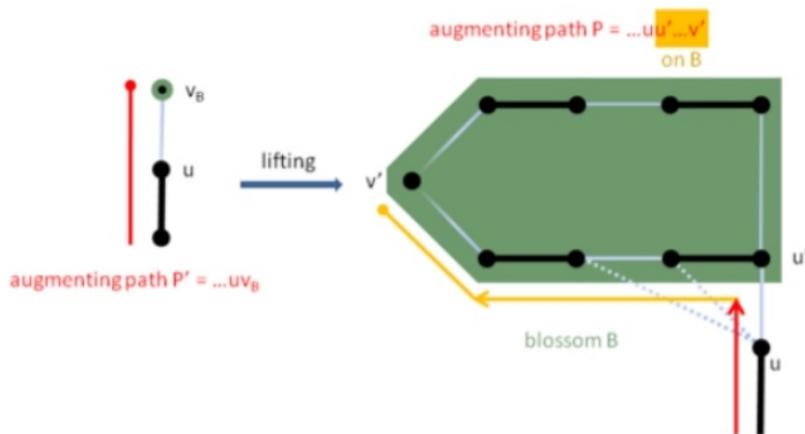
Preuve 2.2. Supposons V_c le sommet de (G') représentant la contraction de (B) soit inclus dans le chemin augmentant (P') pour le couplage (C') , et que ce sommet soit V_B .

1. Si V_B est couvert par C' , alors on a la Fig-37



– Fig-37

2. Si V_B n'est pas couvert par C' alors, on a la Fig-38



– Fig-38

Si le chemin P' est un chemin augmentant pour le couplage C' dans le graphe G , alors il existe également un chemin augmentant pour le couplage C dans le même graphe. Et réciproquement.

Arbre alterné : [1]

Considérons C un couplage dans un graphe G .

Un arbre T dans un graphe G est considéré comme un **"arbre alterné"** par rapport

à un couplage C donné, si et seulement si les deux conditions suivantes sont satisfaites simultanément :

1. Il existe un unique sommet r dans T qui n'est pas couvert par le couplage C . Ce sommet r est considéré comme la racine de l'arbre alterné T .
2. Pour tout autre sommet x appartenant à T et couvert par C , le chemin reliant x à la racine r dans T est un "chemin alternant", c'est-à-dire composé d'arêtes qui alternent entre celles incluses dans C et celles exclues de C .

On définit également les conventions suivantes sur les sommets de T :

- Un sommet x est dit "interne" si sa distance à la racine r (longueur du chemin de r à x) est un nombre impair.
- Un sommet x est dit "externe" si sa distance à la racine r est un nombre pair.

Remarque sur les arbres alternés :

Concernant ces arbres alternés, on peut énoncer les remarques suivantes :

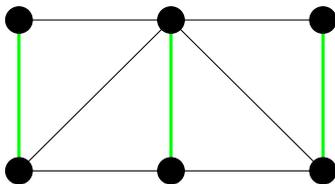
Remarque 2.3.2. *Si un sommet x de T est adjacent dans G à un autre sommet y de même type (interne ou externe) que x , alors en ajoutant l'arête (x,y) au chemin de x à y dans T , on forme un "blossom" (cycle impair).*

Remarque 2.3.3. *Si un sommet x de T est adjacent à un sommet y dans G où y n'appartient pas à T et n'est pas couvert dans C , alors en ajoutant l'arête (x,y) au chemin de la racine r à x dans T , on forme un "chemin augmentant" pour le couplage C .*

2.3.3 Couplage parfait

Un couplage parfait dans un graphe simple non orienté est un ensemble d'arêtes tel que chaque sommet du graphe est relié à exactement une arête du couplage.

Exemple 23.



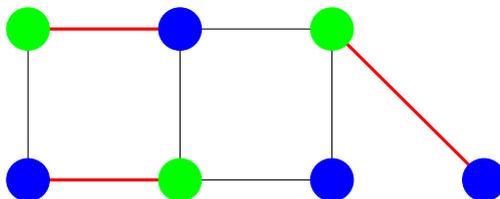
– Fig-39

— En vert, un couplage parfait est de cardinalité $|C| = 3$

2.3.4 Couplage dans graphe biparti

Un couplage dans un graphe biparti est un ensemble d'arêtes qui ne partagent pas de sommets communs.

Exemple 24.



– Fig-40

— En rouge, un couplage dans un graphe biparti est de cardinalité $|C| = 3$

L'affectation maximal ou minimal est aussi un couplage dans un graphe biparti :

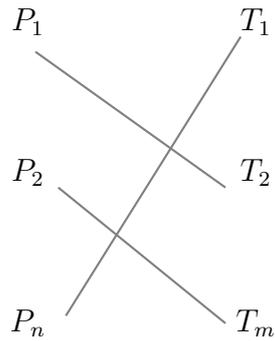
Le problème d'affectation dans un graphe biparti :

Position de problème :

Soit $G = (X, E)$ un graphe avec $X = \{P \cup T\}$ l'ensemble des personnes et des tâches.

$U = \{(p_i, t_j) / i = 1, \dots, n, j = 1, \dots, m\}$.

L'ensemble des arêtes reliant les sommets de P aux sommets de T :



-G-

- Fig-41

La matrice d'affectation associée à ce problème est :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

a_{ij} : est la valeur portée sur l'arrête (P_i, T_j)

Définition 2.3. *les zéros indépendants sont des zéros n'appartient ni à la même ligne, ni à la même colonne.*

Résolution du problème d'affectation minimale :

Méthode hongroise : soit $A = (a_{ij})$ une matrice d'affectation de type (n, n) .

Etapas 1 :

1. Pour créer des zéros sur chaque ligne et chaque colonne de la matrice, soustrayez le plus petit élément de chaque ligne, puis répétez le processus pour chaque colonne.
2. Pour chaque ligne, sélectionnez un zéro et encadrez-le, puis barrez tous les autres zéros de la même ligne et de la même colonne.
3. Si l'on a trouvé n zéros indépendants (encadrés), alors l'affectation est optimale. Chaque zéro encadré correspondants a la valeur a_{ij} dans la matrice d'affectation alors la personne P_i est affectation minimale est égale à la somme des a_{ij} de la matrice de

départ

$$V(aff) = \sum a_{ij}, i = \overline{1, n}, j = \overline{1, m}$$

Si non on passe a la deuxième étape :

1. Marquer toutes les lignes qui ne contiennent aucun zéro encadré.
2. Marquer toutes les colonnes qui contiennent un zéro barré dans une ligne marquée.
3. Marquer toutes les lignes qui contiennent un zéro encadré dans une colonne marquée.
4. Revenir a (2) et reprendre la procédure jusqu'à ce que ne pouvoir plus marquer ni de lignes, ni de colonnes.
5. Tracer une barre sur toutes les lignes non marquées et sur toutes les colonnes marquées.
6. On considère les éléments non barrés de la matrice soit m le plus petit élément d'entre eux
 - A chaque élément non barré de la matrice on retranche m
 - A chaque élément barré deux fois, on ajoute m
 - On laisse inchangé les éléments barrés une seule fois
7. On intègre les nouveaux éléments calculés dans la matrice A on obtient une nouvelle matrice.
8. On choisit des nouveaux zéros pour les lignes ne contenant pas de zéros encadrés
9. Si on a n zéros indépendants (encadrés)
 - Aller a (3) de la 1^{re} étape.
 - Si non aller a (2) de 2^{me} étape.

Remarque 2.3.4.

si $n < m$ on ajoute $(m-n)$ lignes nulles.

Si $n > m$ on ajoute $(n-m)$ colonnes nulles.

Exemple 25.

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 4 & 5 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{matrix} -1 \\ -1 \\ -1 \end{matrix} \rightarrow \begin{pmatrix} 1 & 2 & \boxed{0} \\ 3 & 4 & \emptyset \\ \emptyset & \boxed{0} & 1 \end{pmatrix}$$

On a deux zéros encadré ($2 \neq 3$) donc aller à l'étapes 2

Soit $T = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ les éléments non barrée de A .

$$m = \min(1, 2, 3, 4) = 1, m = 1$$

$$A = \begin{pmatrix} \boxed{0} & 1 & \emptyset \\ 2 & 3 & \boxed{0} \\ \emptyset & \boxed{0} & 2 \end{pmatrix}$$

On a trois zéros en cadrés donc la solution est optimale. Donc

$$P_1 \rightarrow T_1$$

$$P_2 \rightarrow T_3$$

$$P_3 \rightarrow T_2$$

La valeur de l'affectation minimale est : $V_{\min}(aff) = 2 + 1 + 1 = 4$

Algorithme de recherche d'une affectation maximale :

1. on repère le grand élément de la matrice A.

Soit M ce nombre, on retransche tous éléments de A de M, on obtient une nouvelle matrice $B = (b_{ij})$

$$\text{Tel que } B_{ij} = \begin{cases} M - a_{ij} & \text{si } a_{ij} \neq \infty \\ \infty & \text{si } a_{ij} = \infty \end{cases}$$

2. si $m = n$ aller a (3)

Si $m \neq n$ on complète la matrice B par des lignes ou des colonnes nulles aller en (3)

3. on applique l'algorithme de l'affectation minimale sur la matrice B. La valeur de l'affectation maximale est égale à la somme de a_{ij} de la matrice de départ A.

Remarque 2.3.5. Si $a_{ij} = \infty$ veut dire que l'affectation de la personne P_i a la tache de T_j est interdite (impossible).

Exemple 26. trouver l'affectation maximale :

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\text{Max}(a_{ij}) = 3$$

Donc

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix}$$

On applique l'algorithme de l'affectation minimale

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix} \begin{matrix} -0 \\ -0 \\ -3 \end{matrix} \rightarrow \begin{pmatrix} 2 & 1 & \boxed{0} \\ \boxed{0} & 2 & 3 \\ \emptyset & \boxed{0} & \emptyset \end{pmatrix}$$

Donc la solution de l'affectation maximale est :

$$P_1 \xrightarrow{3} T_3$$

$$P_2 \xrightarrow{3} T_1$$

$$P_3 \xrightarrow{0} T_2$$

$$V_{\max}(aff) = 3 + 3 + 0 = 6$$

La personne P_3 est rejetée.

Le théorème de konig 1931 : [2]

Soient ;

$\beta'(G)$: cardinal minimum d'une arête-recouvrement du graphe G.

$\beta(G)$: nombre de transversalité du graphe G

$\alpha'(G)$: cardinal maximum d'un couplage d'un graphe G.

$\alpha(G)$: nombre de stabilité du graphe G.

Alors dans un graphe G on a : $\alpha'(G) = \beta(G)$.

Si de plus G est sans sommet isolé : $\alpha(G) = \beta(G)$.

Le théorème de Hall , aussi appelle théorème de mariage , donne une condition nécessaire et suffisante pour qu'un graphe biparti admette un couplage parfait .

Théorème de HALL :

soit $G = (V_1 \cup V_2 : E)$ un graphe biparti sans sommet isolé . Alors :

1. G admet un couplage de V_1 si et seulement si

$$\forall X \subset V_1 : |G(X)| \geq |X|. \quad (2.1)$$

2. G admet un couplage parfait (i.e. un couplage de V) si et seulement si :

$$\forall X \subset V \text{ tel que } X \subset V_1 \text{ ou } X \subset V_2 : |G(X)| \geq |X|. \quad (2.2)$$

Démonstration :

Le 2 est une conséquence de 1 : si G admet un couplage parfait C , alors ce même couplage est évidemment un couplage de V_1 et V_2 . Par application à V_1 et V_2 on obtient (2.2)

Réciproquement, supposons que G vérifie (2.2) : alors G admet un couplage C_1 de V_1 et un couplage C_2 de V_2 , on a conjointement

$$|V_1| - |C_1| \leq |V_2| \text{ et } |V_2| - |C_2| \leq |V_1| \text{ donc}$$

$$|C_1| = |V_2| \text{ et tout sommet de } V_2 \text{ est incident à } C_1 : C_1 \text{ est un couplage de } G$$

la condition (2.1) est clairement nécessaire : soit C un couplage de V_1 et X un sous-ensemble de V_1 ; notons $E(X)$ l'ensemble des arêtes de C qui sont incidentes à un sommet de X ; les extrémités des arêtes de $E(X)$ qui sont dans V_2 forment un sous-ensemble de $G(X)$, qui est de cardinal $|X|$ (car les arêtes de C ne sont pas incidentes), Donc $|G(X)| \geq |X|$.

La condition 2.1 est suffisante. On utilisant la théorème de König

Soit $X = X_1 \cup X_2$ un transversal minimum de G avec $X_1 \subset V_1$ et $X_2 \subset V_2$: d'après le théorème de König , $|X|$ est la cardinal d'un couplage maximum C ; si C n'est pas un couplage de V_1 , c'est que $|C| < |V_1|$, donc $|X| = |X_1| + |X_2| = |C| < |V_1|$ et $|X_2| <$

$|V_1| - |X_1| = |V_1 \setminus X_1|$; comme $|X|$ est un transversal, toute arête est incidente à X_1 ou à X_2 , donc il n'y a aucune arête entre $|V_1 \setminus X_1|$ et $|V_2 \setminus X_2|$; de ce fait toute arête incidente à $V_1 \setminus X_1$ est incidente à V_2 : $G(V_1 \setminus X_1) \leq |X_2| < |V_1 \setminus X_1|$ et (a) n'est pas réalisé.

Corollaire : [3]

Dans un graphe biparti $G(X, Y, E)$ si l'on dénote par $C_0 \subset E$ un couplage et par $T \subset X \cup Y$ un ensemble transversal (c'est-à-dire un ensemble de sommets tel que toute arête de G ait au moins une de ses extrémités dans T) on a :

$$\max|C_0| = \min|T|.$$

En effet, pour C_0 et T données, on a $|T| \geq |C_0|$ (car T admet un point distinct dans chaque arête de C_0 ; puisque C_0 est un couplage). D'autre part, l'ensemble $T = (X - A) \cup T_G(A)$ est tel que chaque arête a au moins l'une de ses extrémités dans T ; c'est donc un ensemble transversal des arêtes de G , et l'on a, d'après le théorème de König :

$$\max|C_0| = \min(|X - A| + |T_G(A)|) \geq \min|T|.$$

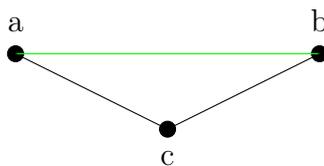
D'où

$$\max|C_0| = \min|T|.$$

2.3.5 Couplage d'un graphe non biparti

Est un ensemble d'arêtes où chaque arête ne partage pas de sommets avec une autre arête du couplage. Il n'y a cependant aucune exigence particulière sur la division des sommets en deux ensembles disjoints.

Exemple 27.



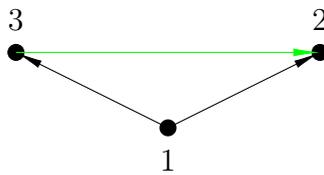
– Fig-42

— En vert, un couplage d'un graphe non biparti est de cardinalité $|C| = 1$

2.3.6 Couplage d'un graphe orienté

Un couplage dans un graphe orienté est un ensemble d'arêtes où chaque arête a une direction spécifique et ne partage pas de sommets avec une autre arête du couplage.

Exemple 28.



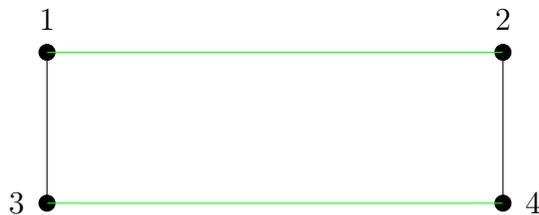
– Fig-43

— En vert, un couplage d'un graphe orienté est de cardinalité $|C| = 1$

2.3.7 Couplage d'un graphe non orienté

Un couplage dans un graphe non orienté est un ensemble d'arêtes où chaque arête ne partage pas de sommets avec une autre arête du couplage, et les arêtes n'ont pas de direction spécifique.

Exemple 29.



– Fig-44

— En vert, un couplage d'un graphe non orienté est de cardinalité $|C| = 2$

2.3.8 Couplage pondéré

Dans le contexte du problème du couplage pondéré, chaque arête d'un graphe est associée à un poids. L'objectif est de trouver un couplage (un ensemble d'arêtes non adjacentes) qui maximise ou minimise la somme des poids des arêtes incluses dans ce couplage. Le choix entre maximisation ou minimisation dépend du problème spécifique que l'on cherche à résoudre.

Exemple 30. *Dans le cas d'un réseau de transport, on peut chercher à maximiser le débit tout en minimisant les coûts associés aux arêtes sélectionnées. Le couplage pondéré est un problème d'optimisation courant en théorie des graphes.*

2.4 Les propriétés du couplage

- Un couplage parfait est nécessairement d'un nombre pair de sommet.
- Un couplage parfait est un couplage de taille maximum.
- Un couplage de taille maximum, n'est pas nécessairement parfait.
- Un couplage parfait est un couplage maximal.
- La taille d'un couplage est le nombre d'arêtes incluses dans ce couplage $|c| =$ le nombre d'arêtes incluses dans ce couplage.

2.5 Les intérêts du couplage

- Affectation des tâches : Les couplages sont employés pour distribuer efficacement les tâches entre des ressources (machines, employés, etc.), tout en minimisant les conflits d'attribution.
- Réseaux de communication : Ils facilitent la planification des liaisons entre les nœuds dans les réseaux de communication, tels que les routeurs, les antennes, etc.
- Optimisation : La recherche d'un couplage maximum (avec le plus grand nombre d'arêtes) est un problème d'optimisation important dans les domaines de la logistique, de la gestion de projets et de la conception des réseaux.

2.6 Conclusion

En conclusion, les problèmes du couplage sont essentiels en théorie des graphes et ont des applications variées dans divers domaines comme l'optimisation, la planification et la gestion des ressources. Ce chapitre a exploré les principaux concepts et méthodes liés aux couplages, y compris les algorithmes et leurs utilisations spécifiques. La maîtrise de ces concepts permet non seulement de mieux comprendre les structures des graphes, mais aussi d'appliquer des solutions efficaces à des problèmes pratiques. Les résultats obtenus démontrent l'importance de ces techniques pour aborder des défis complexes dans les systèmes réels.

Chapitre 3

Complexité des algorithmes

3.1 Introduction

Les algorithmes jouent un rôle essentiel dans la résolution de problèmes complexes, et leur analyse est cruciale pour évaluer leur efficacité. La notation Big O permet de classer les algorithmes selon leur complexité, allant de constante à factorielle. Comparer les performances et les caractéristiques des algorithmes est indispensable pour choisir celui qui est le plus adapté à une tâche spécifique. Par exemple, les algorithmes génétiques offrent certains avantages, mais ils ont aussi des limites, comme la nécessité d'ajuster les paramètres et leur coût élevé en termes des ressources.

3.2 Analyse de complexité temporelle des algorithmes

L'analyse de la complexité temporelle d'un algorithme consiste à évaluer le temps requis pour exécuter cet algorithme en fonction de la taille des données d'entrée. Cette évaluation est cruciale pour comparer l'efficacité de différents algorithmes qui résolvent le même problème.

La complexité temporelle est généralement exprimée à l'aide de la notation Big O, qui indique l'ordre de grandeur de la complexité. Les classes de complexité les plus courantes comprennent :

1. **Complexité constante $O(1)$** : Le temps d'exécution ne dépend pas de la taille

des données d'entrée.

2. **Complexité logarithmique $O(\log(n))$** : Le temps d'exécution croît de manière logarithmique avec la taille des données.
3. **Complexité linéaire $O(n)$** : Le temps d'exécution est directement proportionnel à la taille des données.
4. **Complexité quasi-linéaire $O(n \log(n))$** : Le temps d'exécution croît un peu plus vite que la complexité linéaire.
5. **Complexité quadratique $O(n^2)$** : Le temps d'exécution augmente de manière quadratique avec la taille des données.
6. **Complexité exponentielle $O(2^n)$** : Le temps d'exécution croît de façon exponentielle, devenant vite prohibitif.
7. **Complexité factorielle $O(n!)$** : Une des pires complexités, le temps d'exécution augmente de manière factorielle.

Pour analyser la complexité temporelle d'un algorithme, il est essentiel de prendre en compte le nombre total d'opérations fondamentales effectuées, telles que les comparaisons, les affectations et les accès aux tableaux, ainsi que d'examiner les structures de contrôle comme les boucles et les conditions.

L'analyse de la complexité temporelle est un outil indispensable pour évaluer l'efficacité d'un algorithme et sélectionner le plus approprié pour résoudre un problème donné.

L'algorithme d'Edmonds (1965) :

Complexité : $O(|X|^4)$ car au pire

- on construit $O(|X|)$ chemins augmentants.
- Chaque chemin augmentant est construit en effectuant $O(|X| \times |E|)$ opérations.

Il existe des algorithmes améliorés permettent de construire des chemins en $O(|E|)$ opérations.

3.3 Comparaison de performances et des caractéristiques des algorithmes

La complexité temporelle d'un algorithme mesure son efficacité en termes du temps nécessaire pour accomplir une tâche. Elle est généralement exprimée en utilisant la notation Big O, qui donne une limite supérieure au taux de croissance du temps d'exécution de l'algorithme.

Voici une analyse de la complexité temporelle de quelques algorithmes couramment utilisés :

1. **Recherche d'un maximum dans une liste** : Cette opération consiste à parcourir la liste avec une boucle et à comparer chaque élément avec le maximum actuel. Sa complexité temporelle est $O(n)$, car le nombre d'opérations augmente linéairement avec la taille de la liste.
2. **Recherche d'une valeur dans une liste** : Cette opération implique une boucle qui parcourt la liste et compare chaque élément avec la valeur recherchée. La complexité temporelle est $O(n)$, car le nombre d'opérations croît linéairement avec la taille de la liste.
3. **Tri d'une liste** : Le tri d'une liste peut être réalisé à l'aide d'algorithmes comme le tri rapide ou le tri fusion, qui ont généralement une complexité temporelle de $O(n \log n)$. Ces algorithmes utilisent des structures de contrôle telles que les boucles et les conditions pour organiser les éléments de la liste.
4. **Factorielle** : cet algorithme utilise souvent une boucle qui multiplie les nombres jusqu'à la valeur d'entrée. Sa complexité temporelle est $O(n)$, car le nombre d'opérations augmente linéairement avec la taille de l'entrée.
5. **Recherche binaire** : cet algorithme utilise une approche logarithmique pour trouver une valeur dans une liste triée. Sa complexité temporelle est $O(\log n)$, car le nombre d'opérations augmente logarithmiquement avec la taille de la liste.

Conclusion générale

Cette étude approfondie nous a permis d'explorer en détail la théorie des graphes et le problème fondamental du couplage, ainsi que les algorithmes associés et leur analyse de complexité.

Au premier chapitre, nous avons posé les bases théoriques nécessaires en introduisant les concepts de graphes orientés, non-orientés, l'ordre, le degré des sommets, etc. Ces définitions rigoureuses appuyées par des exemples ont permis d'asseoir les fondations pour la suite.

Le deuxième chapitre a été consacré à la résolution du problème de couplage. Après avoir défini les différents types de couplages (maximal, maximum, parfait, pondéré, etc.) et leurs propriétés, nous avons présenté en détail les principaux algorithmes généraux et spécifiques. L'algorithme d'Edmonds a notamment été expliqué étape par étape. L'algorithme hongrois pour les graphes bipartis pondérés a également été couvert. Des applications concrètes ont illustré l'intérêt pratique des couplages.

Enfin, le troisième chapitre a permis d'analyser facilement la complexité temporelle de ces différents algorithmes. La notation O grande a été introduite, ainsi que les principales classes de complexité (constante, linéaire, quadratique, etc.). Nous avons pu comparer les performances théoriques des algorithmes et identifier leurs forces, faiblesses et compromis en termes de temps de calcul et de qualité de la solution.

Au terme de cette étude, les lecteurs disposent d'une solide compréhension à la fois théorique et pratique du problème de couplage, appuyée par de nombreux exemples,

illustrations et analyses détaillées. Cet ouvrage fournit ainsi les bases conceptuelles et techniques indispensables pour aborder efficacement ce défi central en théorie des graphes et optimisation combinatoire. Les pistes d'améliorations et défis restants identifiés ouvrent également la voie à des futurs progrès dans ce domaine.

Bibliographie

- [1] Couplage Johanne Cohen¹ LLRI-CNRS, Université Paris-Sud, Université Paris-Saclay, France
- [2] Alain Bretto, Alain Faisant, François Hennecart,Éléments de théorie des graphes, chapitre 08, page 249
- [3] Claude Berge,graphes et hypergraphes, deuxième édition,chapitre 07,page 127.

Résumé

Cette thèse explore le problème du couplage en théorie des graphes. Les principaux sujets abordés incluent :

- Les concepts et définitions de base de la théorie des graphes.
- Les différents types de couplage (maximal, maximum, parfait, etc.)
- Les algorithmes pour résoudre les problèmes d'appariement, y compris l'algorithme d'Edmonds et l'algorithme hongrois.
- L'analyse de la complexité temporelle des algorithmes à l'aide de la notation Big O.
- Les comparaisons des performances et des caractéristiques de salgorithmes.

La thèse fournit une compréhension théorique et pratique approfondie des problèmes d'appariement dans les graphes, soutenue par de nombreux exemples et explications détaillées. Elle couvre à la fois les algorithmes généraux et les techniques spécifiques pour les graphes bipartis. L'analyse de la complexité permet de comparer l'efficacité des différentes approches. Dans l'ensemble, ce travail fournit les bases conceptuelles et techniques nécessaires pour aborder efficacement ce défi central en théorie des graphes et en optimisation combinatoire.

Abstract

This thesis explores the matching problem in graph theory. The main topics covered include :

- Basic graph theory concepts and definitions.
- Different types of matchings (maximal, maximum, perfect, etc.).
- Algorithms for solving matching problems, including Edmonds' algorithm and the Hungarian algorithm.
- Analysis of algorithm time complexity using Big O notation.
- Comparisons of algorithm performance and characteristics.

The thesis provides a thorough theoretical and practical understanding of matching problems in graphs, supported by numerous examples and detailed explanations. It covers both general algorithms and specific techniques for bipartite graphs. The complexity analysis allows comparing the efficiency of different approaches. Overall, this work provides the conceptual and technical foundations needed to effectively tackle this central challenge in graph theory and combinatorial optimization.

ملخص

تتناول هذه الأطروحة مشكلة التوفيق في نظرية الرسوم البيانية. وتشمل المواضيع الرئيسية التي تم تغطيتها:

- المفاهيم و التعريفات الأساسية لنظرية الرسوم البيانية.
- الأنواع المختلفة للتوفيق (الأقصى، الأعظم، الكامل، إلخ).
- خوارزميات لحل مشاكل التوفيق، بما في ذلك خوارزمية إدموندز والخوارزمية المجرية.
- Big O تحليل تعقيد الوقت للخوارزميات باستخدام تدوين.
- مقارنات لأداء وخصائص الخوارزميات.

تقدم الأطروحة فهماً نظرياً وعملياً شاملاً لمشاكل التوفيق في الرسوم البيانية، مدعوماً بأمثلة عديدة وشروحات مفصلة. وهي تغطي الخوارزميات العامة والتقنيات الخاصة للرسوم البيانية ثنائية الأطراف. يسمح تحليل التعقيد بمقارنة كفاءة الأساليب المختلفة. بشكل عام، يوفر هذا العمل الأسس المفاهيمية والتقنية اللازمة لمعالجة هذا التحدي المركزي في نظرية الرسوم البيانية والتحسين التوافقي بشكل فعال.