

Introduction à la modélisation

SUPPORT DE COURS

CLASSES PRÉPARATOIRES

par

Fatima Zohra LEBBAH

FZ. LEBBAH

À ma mère

FZ.LEBBAH

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE	8
I INTRODUCTION À LA THÉORIE DES GRAPHS	9
INTRODUCTION	11
1 CONCEPTS DE BASE	12
1.1 Introduction	12
1.2 Graphe orienté	12
1.3 Graphe non-orienté	15
1.4 Quelques définitions de base	17
2 MODES DE REPRÉSENTATION DES GRAPHS	18
2.1 Introduction	18
2.2 Liste de succession	18
2.3 Matrice d'adjacence	19
2.3.1 Graphe orienté	20
2.3.2 Graphe non-orienté	20
2.3.3 Propriétés de la matrice d'adjacence	21
2.4 Matrice d'incidence	21
2.4.1 Graphe orienté	21
2.4.2 Graphe non-orienté	21
3 ÉTUDE DE LA CONNEXITÉ	23
3.1 Introduction	23
3.2 Graphe non orienté	23
3.2.1 Chaîne et chaîne élémentaire	23
3.2.2 Cycle et cycle élémentaire	24
3.3 Graphe orienté	25
3.3.1 Chemin et chemin élémentaire	25
3.3.2 Circuit et circuit élémentaire	26
3.4 Fermeture transitive	27
3.5 Graphe et sous-graphe connexes	31
3.6 Graphe et sous-graphe fortement connexes	33
4 PARCOURS EULÉRIENS ET HAMILTONIENS	37
4.1 Introduction	37
4.2 Parcours Eulériens	37
4.2.1 Origine du problème	37
4.2.2 Chaîne Eulérienne-Cycle Eulérien	38
4.2.3 Graphe Eulérien	39
4.3 Parcours Hamiltonien	40
4.3.1 Origine du problème	40
4.3.2 Chaîne Hamiltonienne-Cycle Hamiltonien	41
4.3.3 Graphe Hamiltonien	42

5	GRAPHES PARTICULIERS	43
5.1	Introduction	43
5.2	Graphe valué	43
5.2.1	Définitions	43
5.3	Réseaux	44
5.4	Graphe complémentaire	45
5.5	Graphe complet	46
5.5.1	Stable	47
5.5.2	Clique	48
5.6	Graphe symétrique	49
5.7	Graphe antisymétrique	50
5.8	Graphe biparti	50
5.9	Graphe planaire	51
5.10	Arbre	52
5.10.1	Terminologie	52
5.10.2	Arbre binaire	53
6	ALGORITHMES DE BASE EN THÉORIE DES GRAPHES	54
6.1	Algorithmes du plus court chemin	54
6.1.1	Graphe à longueurs positives	55
6.1.2	Graphe à longueurs quelconques	57
6.2	Algorithme du flot maximum	59
II INTRODUCTION À LA PROGRAMMATION LINÉAIRE		64
	INTRODUCTION	65
7	RÉSOLUTION D'UN PROGRAMME LINÉAIRE	66
7.1	Introduction	66
7.2	Structure d'un PL	68
7.3	Algorithme du Simplexe	69
7.3.1	Forme du pivot	72
7.3.2	Forme matricielle d'un PL	73
7.4	La dégénérescence et le cyclage	74
8	DUALITÉ	76
8.1	Introduction	76
8.2	Problème dual	76
8.3	Résolution du problème dual	77

TABLE DES FIGURES

FIGURE 1	Graphe orienté $G_{11} = [X_{11}, U_{11}]$	12
FIGURE 2	Graphe $G_{12} = [X_{12}, U_{12}]$ qui modélise la relation « <i>diviseur de</i> » entre les valeurs de l'ensemble $\{1, 2, 3, 4, 5, 6, 7, 8\}$	13
FIGURE 3	Représentation d'un réseau électrique	14
FIGURE 4	Graphe non orienté noté $G_{14} = [X_{14}, U_{14}]$	15
FIGURE 5	Multigraphe $G_{15} = [X_{15}, U_{15}]$	15
FIGURE 6	Carte d'Algérie	16
FIGURE 7	Graphe modélisant la carte d'Algérie coté sud	16
FIGURE 8	Graphe orienté $G_{21} = [X_{21}, U_{21}]$	18
FIGURE 9	Liste de succession correspondant au graphe orienté $G_{21} = [X_{21}, U_{21}]$ de l'exemple 11	19
FIGURE 10	Graphe non-orienté $G_{22} = [X_{22}, U_{22}]$	20
FIGURE 11	Graphe orienté $G_{31} = [X_{31}, U_{31}]$	24
FIGURE 12	Graphe orienté $G_{32} = [X_{32}, U_{32}]$ [Gondran et Minoux, 1995]	26
FIGURE 13	Graphe orienté $G_{33} = [X_{33}, U_{33}]$	28
FIGURE 14	Graphe non-orienté $G_{34} = [X_{34}, U_{34}]$	32
FIGURE 15	Composantes connexes du graphe G_{34} de la figure 14	33
FIGURE 16	Graphe orienté $G_{35} = [X_{35}, U_{35}]$	34
FIGURE 17	Composantes fortement connexes du graphe G_{35} de la figure 16	34
FIGURE 18	Graphe orienté $G_{36} = [X_{36}, U_{36}]$	35
FIGURE 19	Les composantes fortement connexes de G_6 de la figure 18	35
FIGURE 20	Graphe réduit de G_{36} de la figure 18	36
FIGURE 21	Ville de Königsberg [Müller, 2011]	37
FIGURE 22	Exemple de chaîne Eulérienne	38
FIGURE 23	Exemple de cycle Eulérien	38
FIGURE 24	Problème d'Euler modélisé en graphe non orienté $G' = [X', U']$	39
FIGURE 25	Graphes non-orientés : G_{41}, G_{42} et G_{43}	40
FIGURE 26	Le parcours fermé autour du monde par W.Hamilton	41
FIGURE 27	Graphes non-orientés : G_{44}, G_{45} et G_{46}	41
FIGURE 28	Graphe pondéré $G_{51} = [X_{51}, U_{51}]$	44
FIGURE 29	Graphe $G_{52} = [X_{52}, U_{52}]$	45
FIGURE 30	Graphe $G_{53} = [X_{53}, U_{53}]$	45
FIGURE 31	Graphe G_{54} et son complémentaire G_{55} [Hennecart <u>et al.</u> , 2012]	46
FIGURE 32	Graphe complet $G_{56} = [X_{56}, U_{56}]$	46
FIGURE 33	Graphes simples complets K_1, K_2, K_3, K_4 et K_5	47
FIGURE 34	Graphe G_p de Peterson	47
FIGURE 35	Graphe simple $G_{57} = [X_{57}, U_{57}]$	48

FIGURE 36	Graphes G_{58} et \overline{G}_{58}	49
FIGURE 37	Graphe $G_{59} = [X_{59}, U_{59}]$	49
FIGURE 38	Le graphe $G_{510} = [8_{510}, U_{510}]$	50
FIGURE 39	Graphes bipartis	51
FIGURE 40	Arbres dans le graphe $G_{59} = [X_{59}, U_{59}]$	52
FIGURE 41	Arbre $\mathcal{I}' = [X', T']$	53
FIGURE 42	Arbre $\mathcal{I}'' = [X'', T'']$	53
FIGURE 43	$G_{61} = [X_{61}, U_{61}]$ [Gondran et Minoux, 1995]	56
FIGURE 44	Plus courts chemins obtenus sur G_{61} (figure 43)[Gondran et Minoux, 1995]	57
FIGURE 45	$G_{62} = [X_{62}, U_{62}]$ [Gondran et Minoux, 1995]	58
FIGURE 46	Plus courts chemins obtenus sur G_{62} (figure 45)	59
FIGURE 47	$G_{63} = [X_{63}, U_{63}]$	61
FIGURE 48	Flot nul du graphe G_{63}	62
FIGURE 49	Flot φ^1 de G_{63}	63
FIGURE 50	$\overline{G}(\varphi^1)$	63
FIGURE 51	Flot φ^2 de G_{63}	63
FIGURE 52	$\overline{G}(\varphi^2)$	63
FIGURE 53	Flot φ^3 de G_{63}	63
FIGURE 54	$\overline{G}(\varphi^3)$	63
FIGURE 55	Représentation des contraintes de (P_1) (voir le modèle 17)	67
FIGURE 56	Représentation de la fonction objectif de (P_1) (voir le modèle 17)	67
FIGURE 57	Résolution graphique de (P_1) (voir le modèle 17)	68

LISTE DES TABLEAUX

TABLE 1	Table des successeurs	19	
TABLE 2	Table des prédecesseurs	19	
TABLE 3	Table de déroulement de l'algorithme 1 sur le graphe G_{61}	57	
TABLE 4	Table de déroulement de l'algorithme 2 sur le graphe G_{61}	59	
TABLE 5	Table de déroulement de l'algorithme 3 sur le graphe G_{63}	63	
TABLE 6	Table de comparaison entre le problème primal et le problème dual	78	

INTRODUCTION GÉNÉRALE

Au cours de ces dernières années, l'industrie porte un grand intérêt au domaine de la *Recherche Opérationnelle* (RO), une discipline au confluent de l'informatique, des mathématiques appliquées et de l'industrie. La RO peut être conçue comme un ensemble de techniques d'aide à la décision, dont le résultat calculé via des bases rationnelles, représente la meilleure décision que doit prendre l'être humain.

Dans la littérature, nous avons une bibliothèque de solveurs (logiciels de résolution) largement efficaces, tels que GECODE, CPLEX, ILOG SOLVER, GUROBI, BARON, etc.

Une étape de modélisation s'avère indispensable pour avoir une entrée adéquate (un modèle mathématique adéquat) à l'outil de résolution.

Évidemment, l'utilisateur doit avoir la garantie que le modèle conçu reflète réellement le problème posé et que la méthode appliquée offre des résultats numériques corrects et qui représentent effectivement une solution au problème.

Dans ce cours nous apportons une introduction à deux branches de la recherche opérationnelle, à savoir la théorie des graphes et la programmation linéaire. Ce document est conforme au nouveau programme enseigné aux étudiants de la 2^{ème} année des classes préparatoires au sein de notre école. Ce document intitulé « Introduction à la modélisation » fournit, comme son nom l'indique, une introduction aux :

1. concepts fondamentaux de la théorie des graphes et de la programmation linéaire,
2. modélisation d'un problème complexe sous forme d'un modèle graphique ou d'un programme mathématique linéaire,
3. et algorithmes courants, conçus pour la résolution des graphes et des programmes linéaires.

Ce support de cours est organisé de la manière suivante :

- la première partie introduit la théorie des graphes, un domaine qui permet une modélisation robuste d'un problème d'optimisation et qui offre une panoplie d'algorithmes permettant la résolution de problèmes connus dans la littérature, à savoir : le coloriage des graphes, le voyageur de commerce, le plus court chemin, les flots, etc. Le choix de l'algorithme à appliquer est l'étape qui vient après la conception du graphe correspondant à notre problème.

Nous introduisons dans cette partie les notions fondamentales des graphes, quelques graphes particuliers, les principes de modélisation et des algorithmes fréquents.

- la deuxième partie, intitulée « Introduction à la programmation linéaire », apporte les connaissances de base de la modélisation d'un problème d'optimisation sous forme d'un Programme Linéaire (PL)¹, la résolution graphique du *PL*, l'algorithme du *SIMPLEXE* et le principe de la dualité.

1. Un *PL* est un ensemble de contraintes mathématiques d'ordre linéaire

Première partie

INTRODUCTION À LA THÉORIE DES GRAPHS

INTRODUCTION

La théorie des graphes permet une représentation rigoureuse et efficace d'un problème concret, et offre une bibliothèque d'algorithmes qui résolvent un problème modélisé en un graphe. Elle est basée sur trois étapes : (1) modélisation graphique du problème, (2) représentation du graphe correspondant en machine, (3) application d'un algorithme adéquat à la résolution du problème.

La modélisation graphique d'un système physique ou d'un problème complexe s'avère très utilisée dans la résolution des problèmes classés *NP-Comple*t (les problèmes difficiles). La résolution d'un modèle graphique s'avère plus fructueuse et plus prometteuse que la résolution d'un programme mathématique.

Les graphes et leurs algorithmes sont conçus comme des outils mathématiques utilisés pour modéliser et résoudre des problèmes complexes issus de domaines aussi variés que l'optimisation (production industrielle, aide à la décision, etc.), la conception de réseaux (électriques, routiers, télécoms, etc.) ou la modélisation de systèmes évolutifs (économie, automatique, etc.).

Par exemple, on applique :

- un des algorithmes de coloriage des graphes, pour résoudre le problème de coloriage d'une carte géographique, ou du problème de l'emploi du temps, ,
- un des algorithmes de résolution du problème du *plus court chemin*, afin de choisir l'itinéraire le plus court et/ou le plus économique pour traverser d'une ville vers une autre,
- un des algorithmes de résolution du problème du *voyageur du commerce*, afin de choisir le bon chemin pour transmettre une donnée vers tous les postes d'un réseau,
- etc.

L'objet de cette partie est de mettre en avant des techniques fondées sur la théorie des graphes et leur mise en œuvre sur des cas concrets. Chaque concept est muni de graphes et d'exemples illustratifs permettant une meilleure compréhension des principales notions du domaine.

Cette partie « Introduction à la théorie des graphes » regroupe six chapitres :

- Le chapitre 1 donne les concepts de base de la théories des graphes, à savoir les concepts orientés et non-orientés.
- Le chapitre 2 porte sur les techniques de représentation d'un graphe sur machine.
- Le chapitre 3 concerne les principes de l'étude de la connexité dans un graphe.

- Le chapitre 4 introduit les parcours *Eulérien* et *Hamiltonien*.
- Le chapitre 5 représente les propriétés de quelques graphes particuliers, utiles dans la modélisation et le traitement des graphes.
- Le chapitre 6 consacré aux algorithmes de base en théorie des graphes, à savoir le plus court chemin et le flot maximum.

FZ. LEBBAH

CONCEPTS DE BASE

1.1 INTRODUCTION

Dans ce chapitre nous introduisons les définitions et les notions de base de la théorie des graphes, associées d'exemples et de figures illustratives. Dans la littérature nous distinguons deux catégories de graphes :

1. graphe orienté : l'information ou la relation à modéliser n'est pas réciproque entre deux entités, elle est à sens unique.
2. graphe non orienté : l'information ou la donnée à modéliser est réciproque entre deux entités. La liaison entre les deux entités est une relation à deux sens. Par conséquent, le sens de la relation n'est pas modélisé.

1.2 GRAPHE ORIENTÉ

Définition 1 (Graphe orienté) [Gondran et Minoux, 1995] Un graphe $G = [X, U]$ est déterminé par les deux ensembles X et U , tels que :

X : ensemble des éléments appelés sommets (ou nœuds). Si $N = |X|$ est le nombre de sommets (de nœuds), on dit que le graphe G est d'ordre N .

U : ensemble des éléments $u \in U$ qui sont des couples ordonnés de sommets appelés des arcs. Si $u = (i, j)$ est un arc de G , i et j sont respectivement les extrémités initiale et terminale de u . On note par $M = |U|$ le nombre d'arcs.

Pour une meilleure compréhension de la définition 1, nous introduisons ci-dessous deux exemples illustratifs.

Exemple 1 (graphe orienté) Soit le graphe orienté $G_{11} = [X_{11}, U_{11}]$ illustré dans la figure 1.

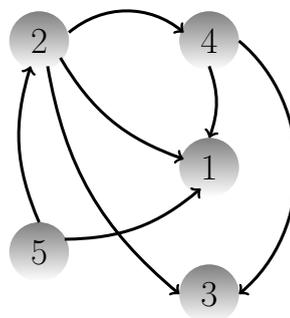


FIGURE 1 – Graphe orienté $G_{11} = [X_{11}, U_{11}]$

On définit le graphe G_{11} par :

— l'ensemble des sommets $X_{11} = \{1, 2, 3, 4, 5\}$

— et l'ensemble des arcs $U_{11} = \{(2, 1), (2, 3), (2, 4), (4, 1), (4, 3), (5, 1), (5, 2)\}$

Ce qui donne :

$N_{11} = 5$: le nombre de sommets,

$M_{11} = 7$: le nombre d'arcs.

Exemple 2 (modélisation de la relation « diviseur de ») Le graphe $G_{12} = [X_{12}, U_{12}]$ de la figure 2 représente (ou modélise) la relation « diviseur de » entre les éléments de l'ensemble $\{1, 2, 3, 4, 5, 6, 7, 8\}$. Nous avons :

$X_{12} = \{1, 2, 3, 4, 5, 6, 7, 8\}$ et

U_{12} = ensemble des arcs, la présence de l'arc (i, j) indique que « i est un diviseur de j ».

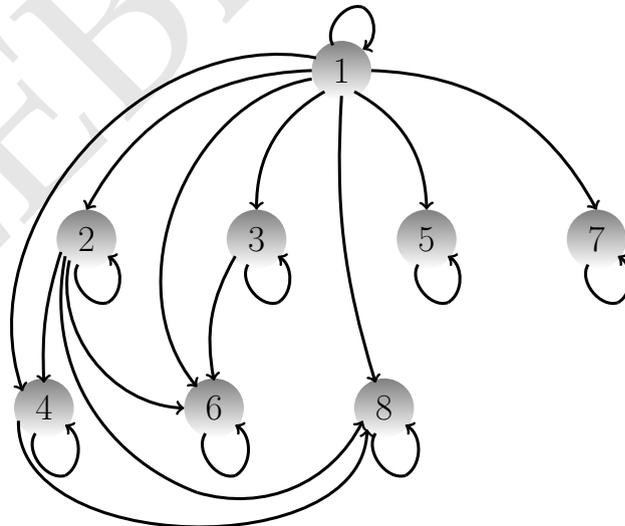


FIGURE 2 – Graphe $G_{12} = [X_{12}, U_{12}]$ qui modélise la relation « diviseur de » entre les valeurs de l'ensemble $\{1, 2, 3, 4, 5, 6, 7, 8\}$

Puisqu'un entier est diviseur de lui-même, nous représentons la relation par une boucle¹.

Exemple 3 (modélisation d'un circuit électrique) [Thulasiraman, 2003] La figure 3 illustre la modélisation du circuit électrique de la sous-figure 3a en un graphe orienté G_{13} donné dans la sous-figure 3b.

Nous avons :

$X_{13} = \{a, b, c, d\}$ et

$U_{13} = \{(a, b), (a, c), (b, d), (b, c), (c, d), (d, a)\}$.

Par exemple l'équation KVL du circuit a, b, d, a et l'équation KCL du sommet b dans le graphe G_{13} se donnent respectivement, comme suit :

1. Une boucle est un arc dont les deux extrémités coïncident.

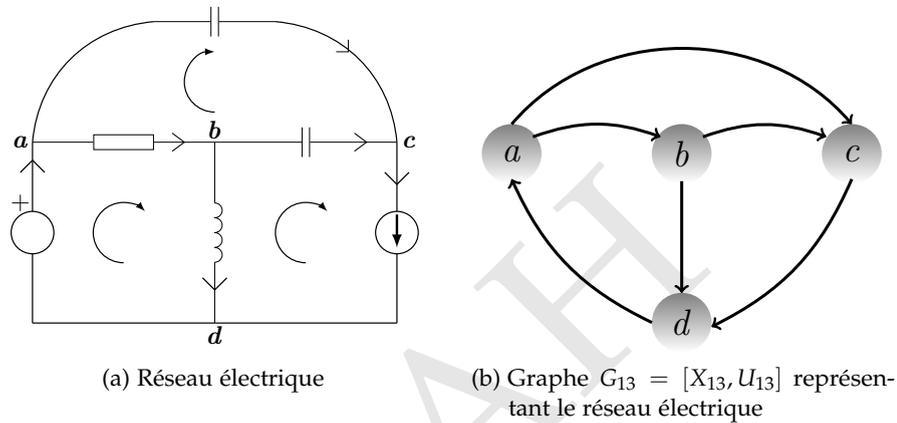


FIGURE 3 – Représentation d'un réseau électrique

le circuit a, b, d, a : $v_{ab} + v_{bd} + v_{da} = 0$

le sommet b : $-i_{ab} + i_{bc} + i_{bd} = 0$

Définition 2 (p-graphe) Un p -graphe est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (i, j) entre deux sommets quelconques i et j , pris dans le même sens.

Exemple 4 (p-graphe) Le graphe illustré dans la figure 1 est un 1-graphe, puisque le maximum d'arcs identiques entre n'importe quelle paire de sommets est 1.

Définition 3 (sommet successeur) On dit que j est un successeur de i s'il existe un arc ayant i comme extrémité initiale et j comme extrémité terminale. L'ensemble des successeurs d'un sommet $i \in X$ est noté par Γ_i

Définition 4 (sommet prédécesseur) On dit que j est un prédécesseur de i s'il existe un arc ayant j comme extrémité initiale et i comme extrémité terminale. L'ensemble des prédécesseur d'un sommet $i \in X$ est noté par Γ_i^{-1} .

Exemple 5 (fonctions Γ et Γ^{-1}) Soit le graphe illustré à la figure 1. On définit les fonctions successeur et prédécesseur comme suit :

- la fonction successeur Γ : $\Gamma_1 = \{\}, \Gamma_2 = \{1, 3, 4\}, \Gamma_3 = \{\}, \Gamma_4 = \{1, 3\}, \Gamma_5 = \{2\}$,
- la fonction prédécesseur Γ^{-1} : $\Gamma_1^{-1} = \{2, 4\}, \Gamma_2^{-1} = \{5\}, \Gamma_3^{-1} = \{2, 4\}, \Gamma_4^{-1} = \{2\}, \Gamma_5^{-1} = \{\}$,

Exemple 6 (fonctions Γ et Γ^{-1}) Soit le graphe illustré dans la figure 2. Nous avons :

$\Gamma_2 = \{2, 4, 8\}$: représente l'ensemble des multiples de 2.

$\Gamma_2^{-1} = \{1, 2\}$: représente l'ensemble des diviseurs de 2.

Après application des deux fonctions sur des sommets du graphe, nous constatons que pour un sommet i :

Γ_i = ensemble des multiples de i .

Γ_i^{-1} = ensemble des diviseurs de i .

1.3 GRAPHE NON-ORIENTÉ

Définition 5 (graphe non-orienté) Un graphe $G = [X, U]$ est déterminé par les ensembles X et U , avec :

X : ensemble des éléments appelés sommets. Si $N = |X|$ est le nombre de sommets, on dit que le graphe G est d'ordre N .

U : ensemble des éléments $u \in U$ qui sont des couples non ordonnés de sommets appelés des arêtes. Si $u = (i, j)$ est une arête de G , i et j sont les extrémités de u . On note par $M = |U|$ le nombre d'arêtes.

Exemple 7 (graphe non orienté) Soit le graphe $G_{14} = [X_{14}, U_{14}]$ donné dans la figure 4, ci-dessous :

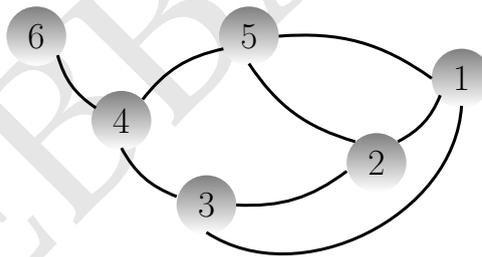


FIGURE 4 – Graphe non orienté noté $G_{14} = [X_{14}, U_{14}]$

On définit le graphe G_{14} par :

- $X_{14} = \{1, 2, 3, 4, 5, 6\}$: ensemble des sommets,
- $U_{14} = \{(1, 2), (1, 3), (1, 5), (2, 3), (2, 5), (3, 4), (4, 5), (4, 6)\}$: ensemble des arêtes.

Ce qui donne :

$N_{14} = 6$: nombre de sommets,

$M_{14} = 8$: nombre d'arêtes.

Définition 6 (Multigraphe) Un multigraphe est un graphe non orienté qui peut contenir plusieurs arêtes entre la même paire de sommets.

Exemple 8 (multigraphe) Soit le graphe $G_{15} = [X_{15}, U_{15}]$ illustré dans la figure 5, ci-dessous :

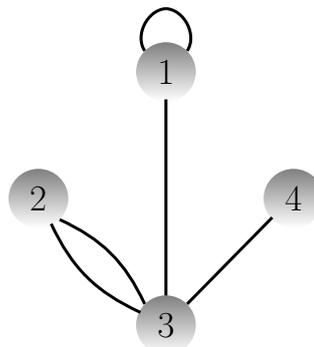


FIGURE 5 – Multigraphe $G_{15} = [X_{15}, U_{15}]$

Le graphe G_{15} ci-dessus est un multigraphe puisqu'il contient deux arêtes entre les deux sommets 2 et 3.

Définition 7 (graphe simple) Un graphe simple est un graphe sans boucle et dont toute paire de sommets est reliée par au plus une arête.

Exemple 9 (graphe simple) Le graphe G_{13} illustré par la figure 4 est un graphe simple puisque chaque paire de sommets est lié par une seule arête au maximum.

Afin de mettre en avant la notion de *non-orienté*, nous proposons ci-dessous un exemple réel d'un graphe non orienté.

Exemple 10 (Modélisation non orientée) La figure 6 illustre la carte d'Algérie. Les wilayas du sud sont données avec leurs codes.

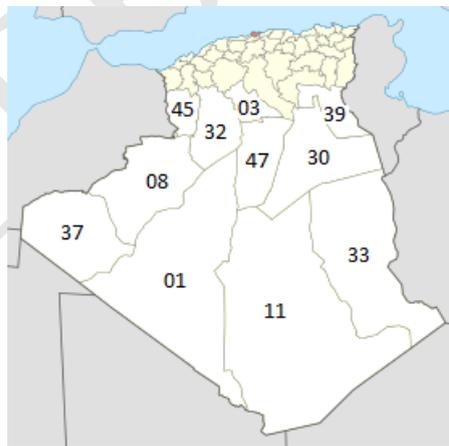


FIGURE 6 – Carte d'Algérie

Le coloriage de la carte nécessite à ce que deux wilayas voisines n'aient pas la même couleur.

La figure 7 représente le graphe modélisant le problème de coloriage de la carte, partie sud, illustrée ci-dessus. Un sommet représente une wilaya et une arête entre deux wilayas v_1 et v_2 représente la relation réciproque « voisin de ».

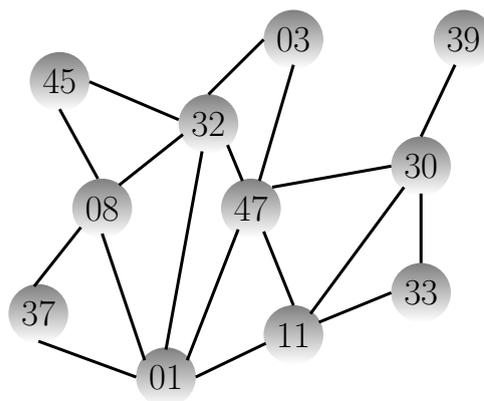


FIGURE 7 – Graphe modélisant la carte d'Algérie côté sud

Après application de l'un des algorithmes de coloriage de graphes nous saurions colorier notre carte avec le minimum de couleurs.

1.4 QUELQUES DÉFINITIONS DE BASE

Définition 8 (arcs adjacents(arêtes adjacentes)) Deux arcs (arêtes) sont dits(es) adjacents(es) s'ils (elles) ont au moins une extrémité commune.

Définition 9 (degré et demi-degrés)

- Le demi-degré extérieur du sommet i , noté $d_G^+(i)$, est le nombre d'arcs ayant i comme extrémité initiale.
- Le demi-degré intérieur du sommet i , noté $d_G^-(i)$, est le nombre d'arcs ayant i comme extrémité terminale.
- Le degré du sommet i , noté $d_G(i)$, est le nombre d'arcs (ou d'arêtes) ayant i comme extrémité, ce qui donne dans le cas orienté²

$$d_G(i) = d_G^+(i) + d_G^-(i) \quad (1)$$

Définition 10 (arcs incidents (arêtes incidentes)) Soit A un sous-ensemble de X , on définit :

$\omega^+(A)$ = ensemble des arcs ayant leurs extrémités initiales dans A et leurs extrémités terminales dans $\bar{A} = X - A$.

$\omega^-(A)$ = ensemble des arcs ayant leurs extrémités terminales dans A et leurs extrémités initiales dans $\bar{A} = X - A$.

On note par $\omega(A) = \omega^+(A) \cup \omega^-(A)$, ensemble d'arcs (ou d'arêtes) appelé un cocycle du graphe.

Définition 11 (graphe partiel) Soit le graphe $G = [X, U]$. Le graphe $G^p = [X^p, U^p]$ est un graphe partiel de G si et seulement si $X^p = X$ et $U^p \subset U$.

Définition 12 (sous-graphe) Soit le graphe $G = [X, U]$. Le graphe $G^s = [X^s, U^s]$ est un sous-graphe de G si et seulement si $X^s \subset X$ et $U^s \subset U$.

2. Si le sommet i comporte une boucle, alors celle ci compte double puisque les deux extrémités sont incidentes à i .

 MODES DE REPRÉSENTATION DES GRAPHS

2.1 INTRODUCTION

Comme nous l'avons mentionné précédemment, la théorie des graphes est un outil puissant de modélisation et de résolution des problèmes difficiles. Généralement, les graphes modélisant un problème complexe, sont des graphes denses et leurs traitements nécessitent une bonne mise en œuvre sur machine.

L'application des algorithmes sur les graphes impose une représentation adaptée sur ordinateur.

Plusieurs modes de représentation peuvent être envisagés. Le choix entre ces différents modes doit être conforme au type de traitements que l'on souhaite appliquer sur le graphe en question.

2.2 LISTE DE SUCCESSION

Considérons le graphe $G = [X, U]$, afin de représenter le graphe G avec la liste de succession nous utilisons :

1. un tableau, appelé *dictionnaire*, qui contient les sommets du graphe G .
2. une liste de successeurs (ou de prédécesseurs) à faire correspondre à chaque élément du dictionnaire.

Exemple 11 (liste de succession) Soit le graphe $G_{21} = [X_{21}, U_{21}]$ illustré dans la figure 8 ci-dessous.

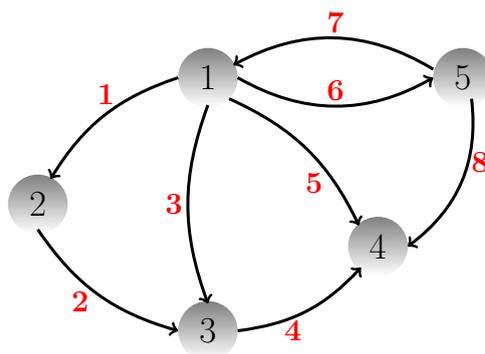


FIGURE 8 – Graphe orienté $G_{21} = [X_{21}, U_{21}]$

Les arcs sont numérotés en rouge de 1 à 8.

Les tables 1 et 2 représentent respectivement les listes de successeurs et de prédécesseurs de chaque sommet :

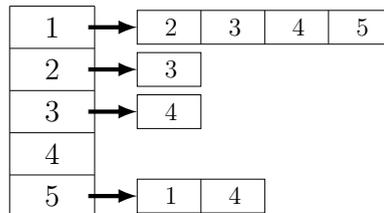
TABLE 1 – Table des successeurs

1	2,3,4,5
2	3
3	4
4	-
5	1,4

TABLE 2 – Table des prédécesseurs

1	5
2	1
3	1,2
4	1,3,5
5	1

La représentation sur machine de la liste de succession ce donne via la figure 9, comme suit :

FIGURE 9 – Liste de succession correspondant au graphe orienté $G_{21} = [X_{21}, U_{21}]$ de l'exemple 11

2.3 MATRICE D'ADJACENCE

Dans ce cas, nous considérons l'arc (i, j) et sa représentation via une matrice carrée booléenne, appelée matrice d'adjacence ou matrice d'incidence sommet-sommet.

Définition 13 (Matrice d'adjacence) *Considérons le graphe $G = [X, U]$ d'ordre N . La matrice d'adjacence de G est égale à la matrice $A = (a_{ij})$ de dimension $N \times N$, telle que :*

$$a_{ij} = \begin{cases} 1 & \text{si } (i, j) \in U \\ 0 & \text{sinon} \end{cases}$$

Comme nous pouvons le constater, la matrice A est une matrice booléenne.

2.3.1 Graphe orienté

La matrice d'adjacence d'un graphe orienté est une matrice quelconque avec des 0 et des 1.

Exemple 12 (Exemple de matrice d'adjacence d'un graphe orienté) Soit le graphe G_{21} de la figure 8 donné dans l'exemple 11.

La matrice d'adjacence A_{21} correspondant à G_{21} est la suivante :

$$A_{21} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

2.3.2 Graphe non-orienté

La matrice d'adjacence d'un graphe non-orienté est une matrice symétrique par rapport à la diagonale.

Exemple 13 (matrice d'adjacence d'un graphe non-orienté) Soit le graphe non-orienté $G_{22} = [X_{22}, U_{22}]$ donné via la figure 10 ci-dessous :

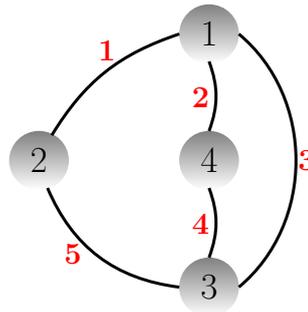


FIGURE 10 – Graphe non-orienté $G_{22} = [X_{22}, U_{22}]$

La matrice d'adjacence correspondant à G_{22} est la suivante :

$$A_{22} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

On remarque que :

- A_{22} est symétrique,
- la diagonale de A_{22} est nulle, puisque le graphe est sans boucle.

2.3.3 Propriétés de la matrice d'adjacence

Considérons le graphe $G = [X, U]$, la matrice d'adjacence A correspondant à G possède les propriétés suivantes :

1. la somme des éléments de la $i^{\text{ème}}$ ligne correspond au demi-degré-extérieur de i dans G , noté $d_G^+(i)$,
2. la somme des éléments de la $j^{\text{ème}}$ colonne correspond au demi-degré-intérieur de j dans G , noté $d_G^-(j)$.
3. A est symétrique si et seulement si le graphe est symétrique ; $d_G^+(i) = d_G^-(i)$ pour tout i dans X .

2.4 MATRICE D'INCIDENCE

Cette représentation exploite la relation sommet-arc (ou sommet arrête).

2.4.1 Graphe orienté

Définition 14 (matrice d'incidence d'un graphe orienté) Considérons le graphe orienté sans boucle $G = [X, U]$ d'ordre N , avec :

$X = \{x_1, \dots, x_N\}$: ensemble des sommets,

$U = \{a_1, \dots, a_M\}$: ensemble des arcs.

On appelle matrice d'incidence de G , la matrice $A = (a_{ij})$ de dimension $N \times M$, telle que :

$$a_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est l'extrémité initiale de } u_j \\ -1 & \text{si } x_i \text{ est l'extrémité terminale de } u_j \\ 0 & \text{si } x_i \text{ n'est pas une extrémité de } u_j \end{cases}$$

Exemple 14 (matrice d'incidence d'un graphe orienté) Soit le graphe G_{21} de l'exemple 11. La matrice d'incidence sommet-arc correspondante se donne comme suit :

$$A_{21} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{bmatrix}$$

2.4.2 Graphe non-orienté

Définition 15 (matrice d'incidence d'un graphe non-orienté) Considérons le graphe non-orienté et sans boucle $G = [X, U]$ d'ordre N , avec :

- $X = \{x_1, \dots, x_N\}$: ensemble des sommets,
- $U = \{a_1, \dots, a_M\}$: ensemble d'arêtes.

On appelle *matrice d'incidence* de G la matrice $A = (a_{ij})$ de dimension $N \times M$, telle que :

$$a_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est l'extrémité de } u_j \\ 0 & \text{sinon} \end{cases}$$

Exemple 15 (matrice d'incidence d'un graphe non-orienté) Soit le graphe G_{22} du graphe illustré dans la figure 10 (exemple 13). La matrice d'incidence sommet-arête correspondant à G_{22} est la suivante :

$$A_{22} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

ÉTUDE DE LA CONNEXITÉ

3.1 INTRODUCTION

Dans ce chapitre nous introduisons la notion de connexité en théorie des graphes. Nous donnons en premier lieu les définitions et les notions de base pour les deux types de graphes, à savoir :

1. chaîne et cycle dans le cas non-orienté,
2. chemin, circuit et fermeture transitive dans le cas orienté.

Puis, nous présentons dans les deux dernières sections, les concepts de connexité et de forte connexité.

3.2 GRAPHE NON ORIENTÉ

3.2.1 Chaîne et chaîne élémentaire

Définition 16 (chaîne) Une chaîne est une séquence finie et alternée de sommets et d'arêtes, débutant et finissant par deux sommets appelés respectivement extrémité initiale et extrémité terminale, telle que chaque arête est incidente avec les sommets qui l'encadrent dans la séquence.

Définition 17 (longueur de la chaîne) La longueur d'une chaîne est égale au nombre d'arêtes qui la composent.

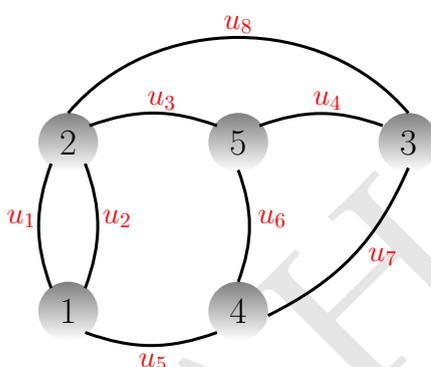
Définition 18 (longueur de la chaîne) [Gondran et Minoux, 1995] Une chaîne de longueur q (ou de cardinalité q) est une séquence de q arêtes :

$$L = \{u_1, u_2, \dots, u_q\}$$

telle que chaque arête u_r de la séquence ($2 \leq r \leq q - 1$) ait une extrémité commune avec l'arête u_{r-1} ($u_{r-1} \neq u_r$) et l'autre extrémité avec l'arête u_{r+1} ($u_{r+1} \neq u_r$).

L'extrémité i de u_1 non adjacente à u_2 et l'extrémité j de u_q non adjacente à u_{q-1} sont appelées respectivement les extrémités initiale et terminale de la chaîne L . Autrement dit, la chaîne L joint les extrémités i et j .

Exemple 16 (chaîne et longueur d'une chaîne) Soit le graphe $G_{31} = [X_{31}, U_{31}]$ de la figure 11 ci-dessous :

FIGURE 11 – Graphe orienté $G_{31} = [X_{31}, U_{31}]$

$L_1 = \{u_8\}$, $L_2 = \{u_1, u_5, u_7\}$, $L_3 = \{u_3, u_4\}$, $L_4 = \{u_1, u_2, u_5, u_7\}$ et $L_5 = \{u_3, u_6, u_7\}$ sont des chaînes qui joignent les sommets 2 et 3. Elles sont de longueurs respectives 1, 3, 2, 4 et 3.

Définition 19 (chaîne élémentaire) [Gondran et Minoux, 1995] Une chaîne élémentaire est une chaîne, telle qu'en la parcourant on ne rencontre pas deux fois le même sommet.

Exemple 17 (chaînes élémentaire et non-élémentaire) Soit le graphe G_{31} de la figure 11. Nous citons parmi les chaînes figurant dans G_{31} :

- $L_1 = \{u_1, u_2, u_3, u_6\}$ est une chaîne qui joint les deux sommets 2 et 4. Elle n'est pas élémentaire puisqu'elle passe plus d'une fois (deux fois) par le sommet 2.
- $L_2 = \{u_6, u_5, u_2\}$ est une chaîne qui joint les deux sommets 5 et 2. Elle est élémentaire puisqu'elle passe une et une seule fois par le même sommet.

Comme nous pouvons le constater, une chaîne est donnée par la suite des arêtes qui la composent. Mais dans le cas d'un graphe non-orienté simple, la chaîne peut être définie par la suite des sommets qui la composent.

Définition 20 (chaîne simple) Une chaîne simple est une chaîne, telle que les arêtes qui la composent sont deux à deux distinctes.

3.2.2 Cycle et cycle élémentaire

Définition 21 (cycle) Un cycle est une chaîne simple dont les deux extrémités coïncident.

Définition 22 (longueur d'un cycle) La longueur d'un cycle est égale au nombre d'arêtes qui le composent.

Définition 23 (cycle et longueur d'un cycle) [Gondran et Minoux, 1995] Un cycle de longueur q est une séquence de q arêtes :

$$\mu = \{u_1, u_2, \dots, u_q\}$$

telle que chaque arête u_r de la séquence ($2 \leq r \leq q - 1$) ait une extrémité commune avec l'arête u_{r-1} ($u_{r-1} \neq u_r$) et l'autre extrémité avec l'arête u_{r+1} ($u_{r+1} \neq u_r$).

L'extrémité i de u_1 non-adjacente à u_2 est l'extrémité de u_q non-adjacente à u_{q-1} .

Définition 24 (cycle élémentaire) [Gondran et Minoux, 1995] Un cycle élémentaire est un cycle minimal (pour l'inclusion) c'est-à-dire ne contenant strictement aucun autre cycle.

En parcourant un cycle élémentaire, on ne rencontre pas deux fois le même sommet.

Exemple 18 (cycles élémentaire et non-élémentaire) Soit le graphe G_{31} de la figure 11. G_{31} contient les deux types de cycle :

- Les cycles $\mu_1 = \{u_1, u_2\}$, $\mu_2 = \{u_1, u_3, u_6, u_5\}$, $\mu_3 = \{u_4, u_6, u_7\}$ et $\mu_4 = \{u_2, u_3, u_4, u_7, u_5\}$ sont de longueurs respectives 2, 4, 3 et 5. Ils sont élémentaires, puisqu'ils ne passent pas plus d'une fois par les sommets qui les composent.
- Les cycles $\mu_5 = \{u_1, u_3, u_6, u_5, u_2\}$ et $\mu_6 = \{u_3, u_2, u_1, u_8, u_4\}$ sont tous les deux de longueur 5. Ils ne sont pas élémentaires puisque :
 - μ_5 passe plus d'une fois par le sommet 1,
 - μ_6 passe plus d'une fois par le sommet 2.

Dans le cas d'un graphe non-orienté simple, le cycle peut être définie par la suite des sommets qui le composent.

3.3 GRAPHE ORIENTÉ

3.3.1 Chemin et chemin élémentaire

Définition 25 (chemin) Un chemin est une chaîne dont tous les arcs sont orientés dans le même sens.

Définition 26 (longueur de la chaîne) La longueur d'un chemin est égale au nombre d'arcs qui le composent.

Définition 27 (chemin et longueur du chemin) [Gondran et Minoux, 1995] Un chemin P de longueur q (de cardinalité q) est une séquence de q arcs :

$$P = \{u_1, u_2, \dots, u_q\}$$

avec :

$$u_1 = (i_0, i_1)$$

$$u_2 = (i_1, i_2)$$

\vdots

$$u_q = (u_{q-1}, u_q)$$

le sommet i_0 est l'extrémité initiale de P ,

le sommet i_q est l'extrémité terminale de P .

Définition 28 (chemin élémentaire) Un chemin élémentaire est un chemin qui passe une et une seule fois par chaque sommet.

Définition 29 (chemin élémentaire) [Gondran et Minoux, 1995] Un chemin élémentaire est un chemin tel qu'on le parcourant, on ne rencontre pas deux fois le même sommet. Dans un chemin élémentaire tous les sommets sont de degré 2 au plus.

Exemple 19 (chemin et chemin élémentaire) Soit le graphe $G_{32} = [X_{32}, U_{32}]$ de la figure 12 ci-dessous :

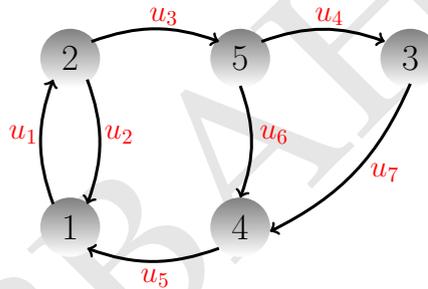


FIGURE 12 – Graphe orienté $G_{32} = [X_{32}, U_{32}]$ [Gondran et Minoux, 1995]

Parmi les chemins contenus dans G_{32} , on peut citer les chemins :

- $P_1 = \{u_1, u_3, u_4, u_7\}$ qui joint sommet 1 au sommet 4. Il est élémentaire de longueur 4.
- $P_2 = \{u_5, u_1, u_2, u_1, u_3\}$ qui joint le sommet 4 au sommet 5. Il est non-élémentaire puisqu'il passe deux fois par les sommets 1 et 2, et de longueur 5.
- $P_3 = \{u_5, u_1, u_3\}$ qui relie le sommet 4 au sommet 5. Il est élémentaire puisqu'il passe une et une seule fois par les sommets qui le constituent et de longueur 3.

Dans le cas d'un 1-graphe, un chemin peut être défini par la séquence des sommets qu'il rencontre dans le même sens.

3.3.2 Circuit et circuit élémentaire

Définition 30 (circuit) [Gondran et Minoux, 1995] Un circuit est un chemin dont les extrémités coïncident.

Définition 31 (longueur d'un circuit) La longueur d'un circuit est égale au nombre d'arcs qui le composent.

Définition 32 (circuit et longueur d'un circuit) [Gondran et Minoux, 1995] Un circuit μ de longueur q (de cardinalité q) est une séquence de q arcs :

$$\mu = \{u_1, u_2, \dots, u_q\}$$

avec :

$$\begin{aligned} u_1 &= (i_0, i_1) \\ u_2 &= (i_1, i_2) \\ &\vdots \\ u_q &= (i_{q-1}, i_0) \end{aligned}$$

Définition 33 (circuit élémentaire) Un circuit est élémentaire si et seulement si en le parcourant on ne passe qu'une et une seule fois par chaque sommet, sauf le sommet choisi comme origine du parcours.

Exemple 20 (circuit et circuit élémentaire) Soit le graphe G_{32} de la figure 12 donnée dans l'exemple ??.

Parmi les circuits ayant le sommet 1 comme origine de parcours, nous avons :

$$\mu_1 = \{u_1, u_2\} \text{ de longueur } 2,$$

$$\mu_2 = \{u_1, u_3, u_6, u_5\} \text{ de longueur } 4,$$

$$\mu_3 = \{u_1, u_3, u_4, u_7, u_5, u_1\} \text{ de longueur } 6,$$

$$\mu_4 = \{u_6, u_5, u_1, u_2, u_3\} \text{ de longueur } 5,$$

...

Les circuits μ_1 , μ_2 et μ_3 sont élémentaires. Par contre le circuit μ_4 ne l'est pas, puisqu'il passe deux fois par les sommets 1 et 2.

3.4 FERMETURE TRANSITIVE

Définition 34 (fermeture transitive d'un sommet) [Gondran et Minoux, 1995] Soient le graphe $G = [X, U]$ d'ordre N et Γ_i^l ensemble des sommets que l'on peut atteindre à partir du sommet i à travers un chemin de longueur l .

On appelle fermeture transitive d'un sommet l'application multivoque $\hat{\Gamma}$ définie par :

$$\hat{\Gamma}_i = \{i\} \cup \Gamma_i^1 \cup \Gamma_i^2 \cup \dots \cup \Gamma_i^{N-1} \quad (2)$$

$\hat{\Gamma}_i$ est l'ensemble des sommets que l'on peut atteindre à partir du sommet i .

On dit que $\hat{\Gamma}_i$ est l'ensemble des descendants de i et $\hat{\Gamma}_i^{-1}$ est l'ensemble des ancêtres de i .

Définition 35 (fermeture transitive d'un graphe) On appelle fermeture transitive d'un graphe $G = [X, U]$, le graphe $G^f = [X, U^f]$ tel que pour toute paire de sommets $(x_i, x_j) \in X^2$, l'arc (x_i, x_j) appartient à U^f si et seulement s'il existe un chemin de x_i vers x_j .

Définition 36 (fermeture transitive d'un graphe) Soient le graphe $G = [X, U]$ d'ordre N et A sa matrice d'adjacence.

On appelle fermeture transitive de G le graphe noté G^f défini par la matrice d'adjacence A^f dont l'élément a_{ij}^f se donne comme suit :

$$a_{ij}^f = \begin{cases} 1 & \text{si } s_{ij} > 0 \\ 0 & \text{si } s_{ij} = 0 \end{cases} \quad (3)$$

tel que s_{ij} est l'élément de la matrice S calculée par 4 :

$$S = \mathcal{I} + A + A^2 + A^3 + \dots + A^{N-1} \quad (4)$$

avec :

\mathcal{I} : matrice identité d'ordre $N \times N$.

Supposons que A est la matrice d'adjacence du graphe $G = [X, U]$. Afin de calculer la fermeture transitive de G notée G^f , on calcule les puissances successives de A , puis les additionner. Le principe de base est simple et nous l'introduisons comme suit :

1. Calcul des puissances¹ de la matrice A :

A : $a_{ij} = 1$ s'il existe un chemin de longueur 1 joignant x_i à x_j ,

A^2 : $a_{ij}^2 = nc_2$ s'il existe nc_2 chemins de longueur 2 joignant x_i à x_j ,

⋮

A^{N-1} : $a_{ij}^{N-1} = nc_{N-1}$ s'il existe nc_{N-1} chemins de longueur $N - 1$ joignant x_i à x_j ,

2. Addition des puissances de A :

A : donne les chemins de longueur 1,

$A + A^2$: donne les chemins de longueur ≤ 2 ,

⋮

$A + A^2 + \dots + A^{N-1}$: donne les chemins de longueur $\leq N - 1$,

Nous pouvons obtenir directement la matrice d'adjacence A^f , en remplaçant le calcul arithmétique par un calcul logique. Ce qui donne la relation logique 5 ci-dessous :

$$A^f = \mathcal{I} \vee A \vee (A \wedge A) \vee (A \wedge A \wedge A) + \dots \underbrace{(A \wedge A \wedge A \wedge \dots)}_{n \text{ fois}} \quad (5)$$

Exemple 21 (fermeture transitive) Considérons le graphe G_{33} de la figure 13.

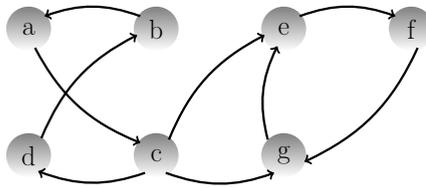


FIGURE 13 – Graphe orienté $G_{33} = [X_{33}, U_{33}]$

Le calcul de la fermeture transitive du graphe G_{33} peut se faire en additionnant les "puissances" successives de sa matrice d'adjacence A comme suit :

1. La matrice d'adjacence A :

1. a_{ij}^k est le coefficient de la ligne i et la colonne j de la matrice A^k

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

2. Calcul de A^2 :

Sachant que :

- $a_{ij} = 1$ s'il existe un chemin de longueur 1 de i vers j ,
- $a_{ik} = 1 \wedge a_{kj} = 1$ implique l'existence d'un chemin de longueur 2 de i vers j ,

Donc, si nous multiplions la matrice A par elle-même nous obtenons les chemins de longueur 2.

$$A^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Pour trouver les chemins de longueur ≤ 2 , nous avons :

- $a_{ij} = 1$ s'il existe un chemin de longueur 1 de i vers j ,
- $a_{ij} = 1 \wedge a_{ij}^2 = nc_2 \geq 1$ implique l'existence de $1 + nc_2$ chemins de longueur ≤ 2 allant de i vers j ,

En additionnant les matrices $A_{33} + A_{33}^2$, nous obtenons les chemins de longueurs 1 ou 2.

$$A_{33} + A_{33}^2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

On appelle $A_{33} + A_{33}^2$ matrice des chemins de longueur inférieure ou égale à 2.

3. La matrice A^3 :

On calcule la matrice A^3 qui fournit les chemins de longueur 3, puis la matrice des chemins de longueur ≤ 3 :

$$A + A^2 + A^3 = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$(A + A^2 + A^3) [1] [5] = 2$ (ou $a_{52} + a_{52}^2 + a_{52}^3 = 2$) signifie que le graphe G_{33} comporte 2 chemins de longueur ≤ 3 joignant 1 à 5.

4. Calcul de A^4 :

On calcule la matrice A^4 qui fournit les chemins de longueur 4, puis la matrice des chemins de longueur ≤ 4 :

$$A + A^2 + A^3 + A^4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 3 & 2 & 3 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 \end{bmatrix}$$

5. Calcul de A^5 :

On calcule la matrice A^5 qui fournit les chemins de longueur 5, puis la matrice des chemins de longueur ≤ 5 :

$$A + A^2 + \dots + A^5 = \begin{bmatrix} 1 & 1 & 2 & 1 & 3 & 2 & 3 \\ 2 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 2 & 5 & 3 & 4 \\ 1 & 2 & 1 & 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 1 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 1 \end{bmatrix}$$

6. Après calcul de A^6 :

Nous avons :

$$|X_{33}| = 7$$

et

$$\mathcal{I} + A + A^2 + \dots + A^6 = \begin{bmatrix} 2 & 1 & 2 & 2 & 5 & 3 & 4 \\ 2 & 2 & 2 & 1 & 3 & 2 & 3 \\ 1 & 2 & 2 & 2 & 6 & 5 & 5 \\ 2 & 2 & 1 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 3 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 3 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 3 \end{bmatrix}$$

Donc, d'après la définition 36, la matrice d'adjacence A^f du graphe G_{33}^f représentant la fermeture transitive de G_{33} , se définit comme suit :

$$A^f = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

3.5 GRAPHE ET SOUS-GRAPHE CONNEXES

Définition 37 (graphe connexe) Soit le graphe non orienté $G = [X, U]$, on dit que le graphe G est connexe si pour toute paire de sommets $(x_i, x_j) \in X^2$, il existe une chaîne joignant x_i à x_j .

Définition 38 (graphe connexe) [Berge, 1985] Un graphe $G = [X, U]$ est connexe si $\forall x \in X$ et $\forall y \in X$ il existe une chaîne de x à y .

Définition 39 (composante connexe) Une composante connexe est un sous-graphe induit maximal $G_i [X_i, U_i] / i \in 1..p$ de G , tel que pour toute paire de sommets $(x, y) \in X_i^2$, il existe une chaîne joignant x et y dans G_i .

Dans [Gondran et Minoux, 1995] la connexité se donne via la relation \mathcal{R} comme suit :

$$x_i \mathcal{R} x_j \Leftrightarrow \begin{cases} x_i = x_j \\ \vee \\ \exists \text{ une chaîne joignant } x_i \text{ à } x_j \end{cases} \quad (6)$$

\mathcal{R} est une relation d'équivalence (réflexivité, symétrie, transitivité).

Les classes d'équivalence induites sur X par \mathcal{R} (voir l'équivalence 6) forment une partition de X en : X_1, X_2, \dots, X_p . En d'autres termes, les sommets contenus dans X_i sont liés deux à deux par la relation \mathcal{R} .

Les sous-graphes $G_i / i \in 1..p$ engendrés par les sous-ensembles de $X_i / i \in 1 \dots p$ sont les composantes connexes du graphe G .

Définition 40 (nombre de connexité) Le nombre de connexité d'un graphe non orienté G est le nombre de composantes connexes contenues dans G , noté p .

A partir de la définition ci-dessous 40, nous concluons qu'un graphe est connexe si et seulement si son nombre de connexité $p = 1$.

Définition 41 (point d'articulation) [Gondran et Minoux, 1995] Un point d'articulation d'un graphe est un sommet dont la suppression augmente le nombre de composantes connexes.

Définition 42 (isthme) [Gondran et Minoux, 1995] Un isthme est une arête dont la suppression a le même effet qu'un point d'articulation.

Exemple 22 (points d'articulation et isthme) *Considérons le graphe G_{34} de la figure 14 :*

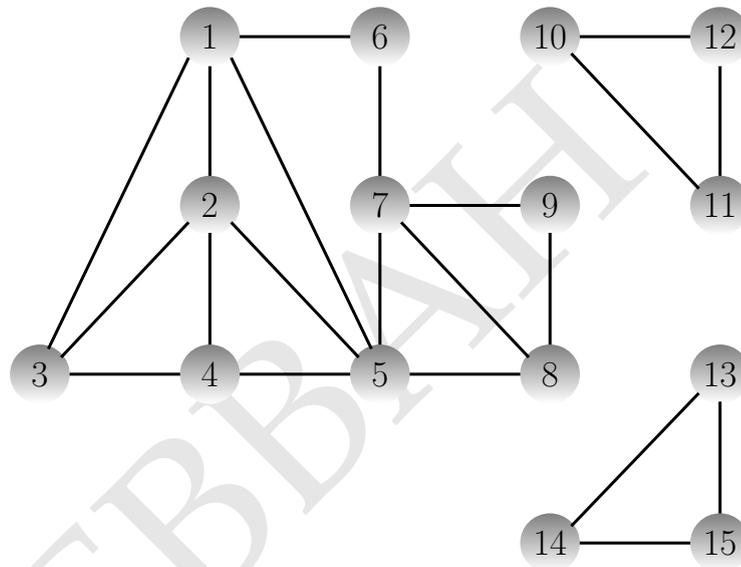
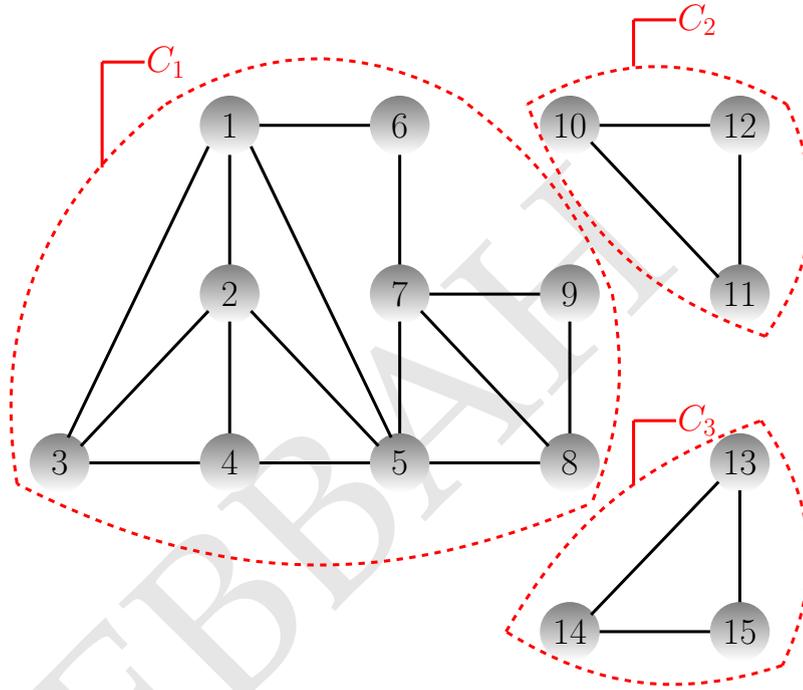


FIGURE 14 – Graphe non-orienté $G_{34} = [X_{34}, U_{34}]$

Le graphe G_{34} n'est pas connexe puisqu'il comporte des paires de sommets non liés par une chaîne, à savoir :

- *entre les sommets de l'ensemble $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ et ceux de l'ensemble $\{13, 14, 15\}$,*
- *entre les sommets de l'ensemble $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ et ceux de l'ensemble $\{10, 11, 12\}$,*
- *entre les sommets de l'ensemble $\{10, 11, 12\}$ et ceux de l'ensemble $\{13, 14, 15\}$.*

Par conséquent, les composantes connexes C_1 , C_2 et C_3 forment des sous graphes connexes, tel illustré dans la figure 15.

FIGURE 15 – Composantes connexes du graphe G_{34} de la figure 14

3.6 GRAPHE ET SOUS-GRAPHE FORTEMENT CONNEXES

Définition 43 (graphe fortement connexe) Soit le graphe orienté $G = [X, U]$, on dit que le graphe G est fortement connexe si pour toute paire de sommets $(x_i, x_j) \in X^2$, il existe deux chemins joignant respectivement x_i à x_j et x_j à x_i .

Définition 44 (composante fortement connexe) Une composante fortement connexe est un sous-graphe induit maximal $G_i [X_i, U_i] / i \in 1..q$ de $G = [X, U]$, tel que pour toute paire de sommets $(x, y) \in X_i^2$, il existe deux chemins joignant respectivement x à y et y à x dans G_i .

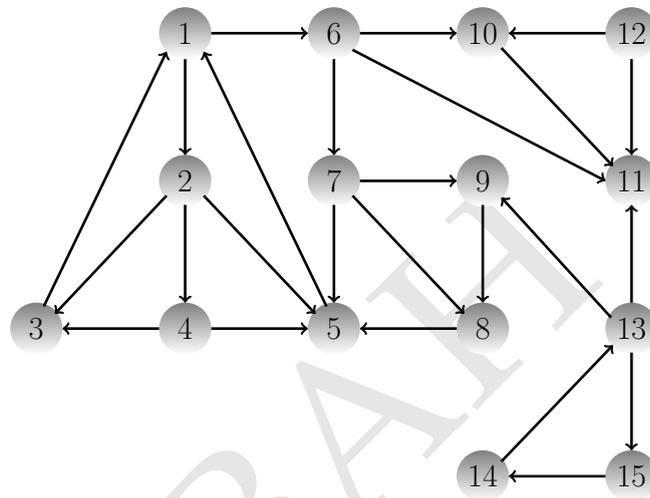
Dans [Gondran et Minoux, 1995] la forte connexité se donne via la relation \mathcal{R} comme suit :

$$x_i \mathcal{R} x_j \Leftrightarrow \begin{cases} x_i = x_j \\ \vee \\ \exists \text{ un chemin joignant } x_i \text{ à } x_j \wedge \exists \text{ un chemin joignant } x_j \text{ à } x_i \end{cases} \quad (7)$$

\mathcal{R} définie par 7 est une relation d'équivalence et les classes d'équivalence induites sur X sont des partitions de X ; $X = X_1 \cup X_2 \cup \dots \cup X_q$.

Les sous-graphes engendrés par $X_i / i \in 1..q$ sont appelés des composantes fortement connexes.

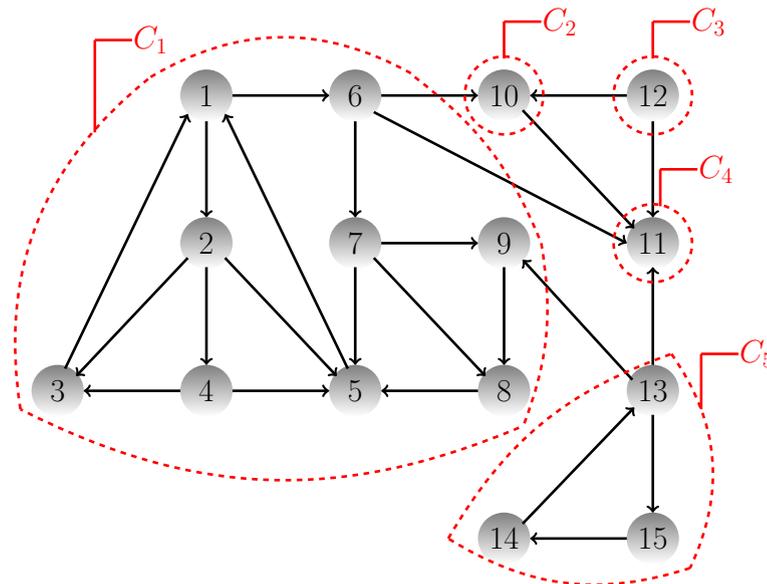
Exemple 23 (sous-graphe ou composante fortement connexe) Considérons le graphe G_{35} de la figure 16 ci-dessous :

FIGURE 16 – Graphe orienté $G_{35} = [X_{35}, U_{35}]$

Le graphe G_{35} n'est pas fortement connexe puisqu'il comporte des paires de sommets non liés par un chemin, à savoir :

- entre les sommets de l'ensemble $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ et ceux de l'ensemble $\{13, 14, 15\}$,
- entre les sommets de l'ensemble $\{10, 11, 12\}$ et ceux de l'ensemble $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$,
- entre les sommets de l'ensemble $\{10, 11, 12\}$ et ceux de l'ensemble $\{13, 14, 15\}$.

Par conséquent les composantes fortement connexes C_1, C_2, C_3, C_4 et C_5 , tel illustré dans 17, forment des sous graphes fortement connexes.

FIGURE 17 – Composantes fortement connexes du graphe G_{35} de la figure 16

Définition 45 (graphe réduit) Soient le graphe $G = [X, U]$ et l'ensemble des composantes fortement connexes $C = \{C_1, C_2, \dots, C_q\}$.

On appelle graphe réduit $G^r = [X^r, U^r]$ de G , le graphe dont :

- $X^r = C$
- et $(C_i, C_j) \in U^r$ si et seulement s'il existe au moins un arc dans G ayant l'extrémité initiale dans C_i et l'extrémité terminale dans C_j

On vérifie que le graphe G^r est sans circuit.

Exemple 24 (graphe réduit) Considérons le graphe $G_{36} = [X_{36}, U_{36}]$ de la figure 18 ci-dessous :

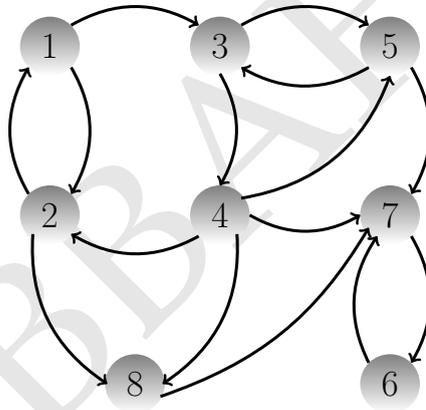


FIGURE 18 – Graphe orienté $G_{36} = [X_{36}, U_{36}]$

Le graphe G_{36} n'est pas fortement connexe puisqu'il comporte des paires de sommets non liés par un chemin, à savoir :

- entre le sommet 8 et les sommets de l'ensemble $\{1, 2, 3, 4, 5\}$,
- entre les sommets de l'ensemble $\{6, 7\}$ et ceux de l'ensemble $\{1, 2, 3, 4, 5\}$,
- entre les sommets de l'ensemble $\{6, 7\}$ et le sommet 8.

Par conséquent les composantes fortement connexes C_1 , C_2 et C_3 illustrées dans la figure 19 forment des sous-graphes fortement connexes.

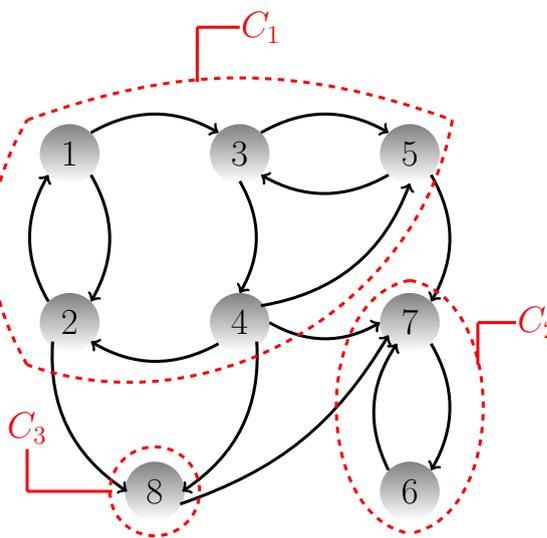


FIGURE 19 – Les composantes fortement connexes de G_6 de la figure 18

Pour obtenir le graphe réduit noté G_{36}^r (voir la figure 20) correspondant à G_{36} :

1. on remplace les composantes fortement connexes par des sommets,
2. et on maintient les arcs qui lient les composantes fortement connexes.

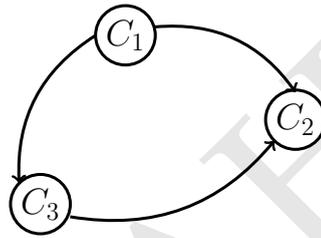


FIGURE 20 – Graphe réduit de G_{36} de la figure 18

PARCOURS EULÉRIENS ET HAMILTONIENS

4.1 INTRODUCTION

Dans ce chapitre nous introduisons le principe et les notions de base des parcours Eulérien et Hamiltonien. Les applications des deux parcours en théorie des graphes s'avèrent nombreuses et très efficaces dans des problèmes d'optimisation de tournées de distribution, tels que le postier chinois et le voyageur de commerce, les problèmes d'ordonnancement, les réseaux d'ordinateurs etc.

4.2 PARCOURS EULÉRIENS

4.2.1 Origine du problème

Le problème de l'existence d'un cycle Eulérien (respectivement d'une chaîne Eulérienne) a été posé pour la première fois et résolu par *Euler* en 1736. Lors d'une promenade dans la ville de *Koenigsberg* traversée par la rivière *Pregel* qui coule de part et d'autre (voir la figure 21). *Euler* s'est posé la question : "Est il possible de passer sur tous les ponts de la ville une et une seule fois et revenir au point de départ ?"

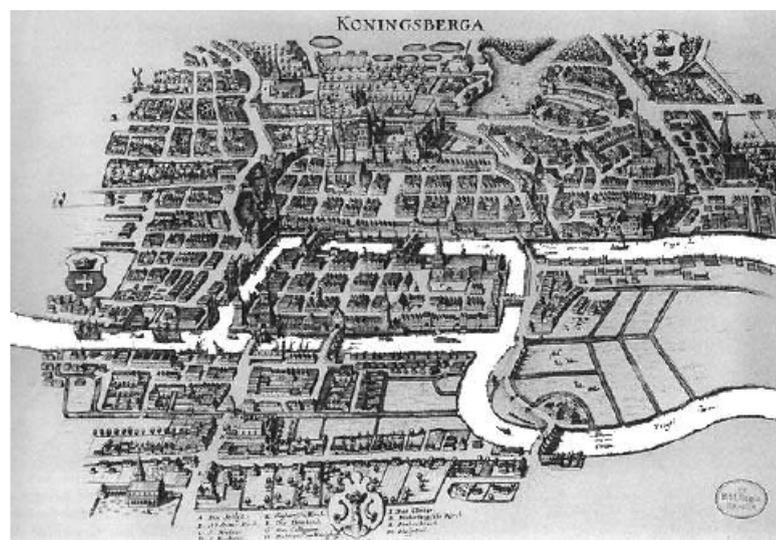


FIGURE 21 – Ville de Königsberg[Müller, 2011]

4.2.2 Chaîne Eulérienne-Cycle Eulérien

Il s'agit d'une généralisation du jeu bien connu consistant à dessiner toutes les arêtes d'un graphe avec un crayon sans jamais le soulever et sans passer deux fois sur la même arête.

Définition 46 (chaîne Eulérienne) [Gondran et Minoux, 1995] Soit le graphe non orienté $G = [X, U]$. Une chaîne Eulérienne est une chaîne empruntant une fois et une fois seulement chaque arête de G .

Exemple 25 (chaîne Eulérienne) Considérons le graphe de la figure 22 ci-dessous :

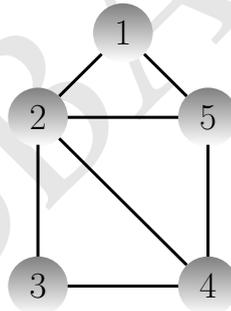


FIGURE 22 – Exemple de chaîne Eulérienne

La chaîne $\{4, 2, 5, 4, 3, 2, 1\}$ est une chaîne Eulérienne puisqu'elle passe une seule fois par chacune des arêtes du graphe.

Définition 47 (cycle Eulérien) [Gondran et Minoux, 1995] Soit le graphe non orienté $G = [X, U]$. Un cycle Eulérien est une chaîne Eulérienne dont les extrémités coïncident.

Définition 48 (cycle Eulérien et chaînes Eulérienne) [Labelle, 1981] Soit le graphe non-orienté $G = [X, U]$. Un cycle Eulérien (respectivement une chaîne Eulérienne) dans G est un cycle (respectivement une chaîne) contenant chaque arête de G une et une seule fois.

Exemple 26 (cycle Eulérien) Considérons le graphe de la figure 23 ci-dessous :

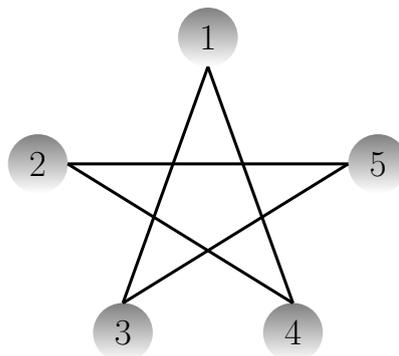


FIGURE 23 – Exemple de cycle Eulérien

Le cycle $\{1, 3, 5, 2, 4, 1\}$ est un cycle Eulérien puisqu'il passe une seule fois par chacune des arêtes du graphe.

4.2.3 Graphe Eulérien

Définition 49 (graphe Eulérien) Soit le graphe non orienté $G = [X, U]$. Le graphe G est dit Eulérien si et seulement s'il contient un cycle Eulérien.

Définition 50 (graphe semi-Eulérien) Soit le graphe non-orienté $G = [X, U]$. Le graphe G est dit semi-Eulérien si et seulement s'il contient une chaîne Eulérienne.

Définition 51 (graphe pré-Eulérien) Soit le graphe non-orienté $G = [X, U]$. Le graphe G est dit pré-Eulérien ou chinois si et seulement s'il contient un cycle empruntant au moins une fois chacune des arêtes de G .

Théorème 1 (théorème d'Euler) Soit le graphe connexe $G = [X, U]$.

- Le graphe G est dit Eulérien si et seulement si tous ses sommets sont de degré pair.
- Le graphe G est dit semi-Eulérien si et seulement si le nombre de sommets de degré impair est 0 ou 2.

Euler prouva, via le théorème 1, l'impossibilité de son problème modélisé par le graphe de la figure 24.

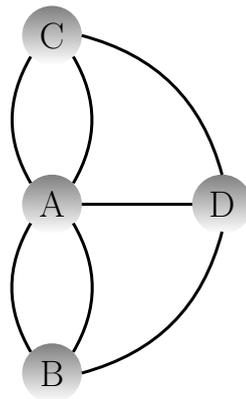


FIGURE 24 – Problème d'Euler modélisé en graphe non orienté $G' = [X', U']$

Dans le graphe G' , nous avons :

$$d_{G'}(A) = 3$$

$$d_{G'}(B) = 5$$

$$d_{G'}(C) = 3$$

$$d_{G'}(D) = 3$$

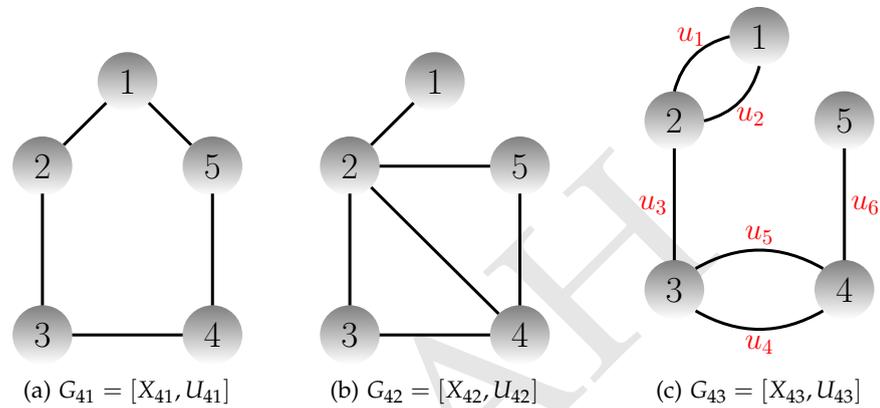
A partir du théorème d'Euler, nous pouvons déduire que le graphe G' :

- n'est pas Eulérien,
- n'est pas Semi-Eulérien.

Par conséquent le problème d'Euler est sans solution.

Exemple 27 (graphes Eulérien, semi-Eulérien et pré-Eulérien) Soient les graphes non-orientés G_{41} , G_{42} et G_{43} illustrés dans la figure 25 ci-dessous :

- G_{41} est Eulérien puisqu'il contient le cycle Eulérien $\{1, 2, 3, 4, 5\}$,

FIGURE 25 – Graphes non-orientés : G_{41} , G_{42} et G_{43}

- G_{42} est semi-Eulérien puisqu'il contient la chaîne Eulérienne $\{1, 2, 3, 4, 5, 2, 4\}$,
- G_{43} est pré-Eulérien puisqu'il contient le cycle $\{u_1, u_2, u_3, u_5, u_6, u_6, u_5, u_3\}$ qui passe par toutes les arêtes du graphe :
 - 1 fois par les arêtes u_1, u_2, u_4, u_5 ,
 - 2 fois par les arêtes u_3, u_6 .

4.3 PARCOURS HAMILTONIEN

4.3.1 Origine du problème

La notion de cycle Hamiltonien trouve son origine dans un jeu inventé par Hamilton en 1859, il s'agit d'un voyage fermé autour du monde.

Tel illustré dans la figure 26 :

- 20 villes réparties sur le globe terrestre sont modélisées par des sommets d'un dodécaèdre régulier représentant la terre,
- les passages directs entre les villes sont représentés par des arêtes.

La question posée : "Est il possible de passer par chacune des villes en utilisant seulement les arêtes du dodécaèdre et de revenir au point de départ ?"

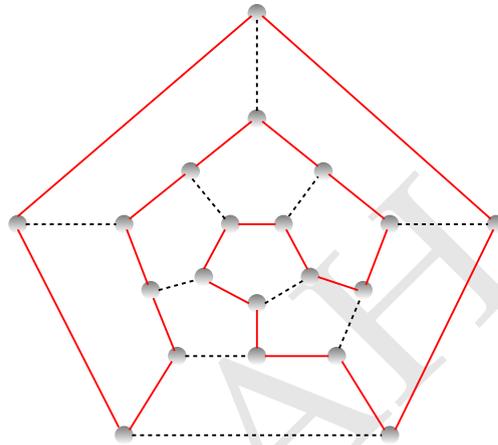


FIGURE 26 – Le parcours fermé autour du monde par W.Hamilton

4.3.2 Chaîne Hamiltonienne-Cycle Hamiltonien

Définition 52 (chaîne Hamiltonienne) [Gondran et Minoux, 1995] Soit le graphe connexe $G = [X, U]$ d'ordre N . On appelle chaîne Hamiltonienne une chaîne passant une fois et une fois seulement par chacun des sommets de G .

Une chaîne Hamiltonienne est donc une chaîne élémentaire de longueur $N - 1$.

Définition 53 (cycle Hamiltonien) [Gondran et Minoux, 1995] Soit le graphe connexe $G = [X, U]$ d'ordre N . On appelle cycle Hamiltonien un cycle passant une fois et une fois seulement par chacun des sommets de G .

Un cycle Hamiltonien est donc un cycle élémentaire de longueur N .

Exemple 28 (cycle Hamiltonien/chaîne Hamiltonienne) Considérons les graphes G_{44} , G_{45} et G_{46} illustrés dans la figure 27 ci-dessous :

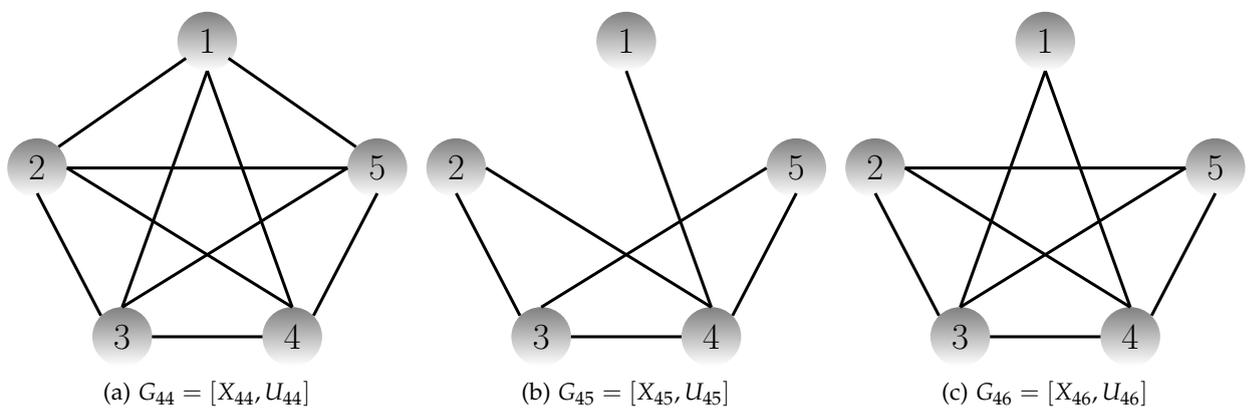


FIGURE 27 – Graphes non-orientés : G_{44} , G_{45} et G_{46}

Nous avons :

- le cycle Hamiltonien $\{1, 2, 3, 4, 5, 1\}$ qui figure dans G_{44} et G_{46} ,
- la chaîne Hamiltonienne $\{1, 4, 2, 3, 5\}$ contenue dans G_{45} .

4.3.3 Graphe Hamiltonien

Définition 54 (graphe Hamiltonien) Soit le graphe connexe $G = [X, U]$ est hamiltonien si et seulement s'il contient un cycle Hamiltonien.

Définition 55 (graphe semi-Hamiltonien) Soit le graphe connexe $G = [X, U]$. Le graphe G est dit semi-Hamiltonien si et seulement s'il contient une chaîne Hamiltonienne.

Définition 56 (graphe pré-hamiltonien) Soit le graphe connexe $G = [X, U]$ sans cycle Hamiltonien. Le graphe G est dit pré-Hamiltonien s'il contient un cycle passant au moins une fois par chacun des sommets de G .

Théorème 2 (théorème de caractérisation de O. Ore) Soit $G = [X, U]$ un graphe simple de degré $N > 2$.

$\forall x, y \in X$ non adjacents, $d_G(x) + d_G(y) \geq N \Rightarrow$ le graphe G est Hamiltonien.

Corollaire 1 (corollaire de Dirac) Soit $G = [X, U]$ un graphe simple d'ordre $N > 2$.
 $\forall x \in X, d_G(x) \geq N/2 \Rightarrow$ le graphe G est Hamiltonien

Exemple 29 (graphe semi/pré/Hamiltonien) Considérons les graphes de la figure 27 donnée dans l'exemple 28

- G_{44} est Hamiltonien puisqu'il contient les cycles Hamiltoniens :
 - $\mu_1 = \{1, 2, 3, 4, 5, 1\}$
 - $\mu_2 = \{1, 4, 2, 5, 3, 1\}$
- G_{45} n'est pas Hamiltonien puisqu'il ne comporte pas de cycle hamiltonien. Il contient la chaîne Hamiltonienne $L_1 = \{1, 4, 2, 3, 5\}$, donc il est semi-Hamiltonien,
- G_{46} n'est pas Hamiltonien puisqu'il ne comporte aucun cycle Hamiltonien. Il contient le cycle non-élémentaire qui passe par tous les sommets du graphe $\mu_3 = \{1, 4, 2, 3, 5, 4, 1\}$, donc il est pré-Hamiltonien.

 GRAPHES PARTICULIERS

5.1 INTRODUCTION

En plus des graphes Eulérien et Hamiltonien décrits précédemment (chapitre 4), nous introduisons dans ce chapitre les définitions et les caractéristiques d'autres graphes particuliers usuels.

L'étude des graphes particuliers et leurs propriétés est indispensable pour l'amélioration d'un algorithme général et ainsi pour le traitement des problèmes qu'ils représentent.

5.2 GRAPHE VALUÉ

5.2.1 Définitions

Définition 57 (graphe valué) Un graphe valué est un graphe $G = [X, U]$ muni d'une fonction de coût : $\gamma : U \rightarrow \mathbb{R}$, tel que $\gamma(u)$ assigne un coût ou une valeur (ou un poids) à l'arc u .

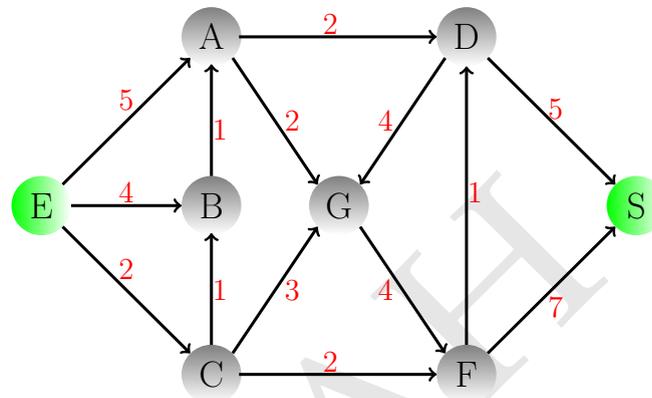
Définition 58 (matrice d'adjacence) Soit le graphe valué $G = [X, U]$. La matrice d'adjacence de G notée A est définie comme suit :

$$a[i][j] = \begin{cases} \gamma((x_i, x_j)) & \text{si } (x_i, x_j) \in U \\ \infty & \text{si } (x_i, x_j) \notin U \end{cases} \quad (8)$$

Définition 59 (coût d'un chemin) Le coût d'un chemin P , noté $\Gamma(P)$ est la somme des valeurs (des poids) de ses arêtes, tel que :

$$\Gamma(P) = \sum_{i=1}^q \gamma(u_i) | u_i \in P, \text{ avec } q : \text{longueur de } P \quad (9)$$

Exemple 30 (graphe valué) Considérons le graphe G_{51} de la figure 28 ci-dessous :

FIGURE 28 – Graphe pondéré $G_{51} = [X_{51}, U_{51}]$

La matrice d'adjacence A correspondant à G_{51} se donne comme suit :

$$A = \begin{bmatrix} \infty & \infty & \infty & 2 & \infty & \infty & 2 & \infty \\ 1 & \infty \\ \infty & 1 & \infty & \infty & \infty & \infty & 2 & 3 \\ \infty & 4 & 5 \\ 5 & 4 & 2 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 1 & \infty & \infty & \infty & \infty & 7 \\ \infty & \infty & \infty & \infty & \infty & \infty & 4 & \infty & \infty \\ \infty & \infty \end{bmatrix}$$

Nous avons ci-dessous le chemin P et son coût (ou son poids) :

$$P = \{E, B, A, G, F, S\}$$

$$\Gamma(P) = 4 + 1 + 2 + 4 + 7 = 18$$

5.3 RÉSEAUX

Définition 60 (réseau) Un réseau est un graphe valué $G = [X, U]$ dans lequel il y a :

- un unique sommet s appelé la source, tel que $d_G^-(s) = 0$,
- et un unique sommet p appelé le puits, tel que $d_G^+(p) = 0$.

Pour $u \in U$, la valeur c_u est appelée la capacité de l'arc u .

La notion de réseaux peut être exploitée pour la modélisation et l'étude des cas réels tels que [Berge, 1985] :

- les circuits électriques où pour $u \in U$, c_u est la capacité du fil électrique u .
- le réseau de transport en commun (en autobus) d'une ville. Les sommets s et p sont les garages d'entretien et c_u est le nombre maximum d'autobus pouvant emprunter l'artère u durant une heure (par exemple).
- ...

Exemple 31 (réseau) [Berge, 1985] Considérons le graphe $G_{52} = [X_{52}, U_{52}]$ de la figure 29 ci-dessous :

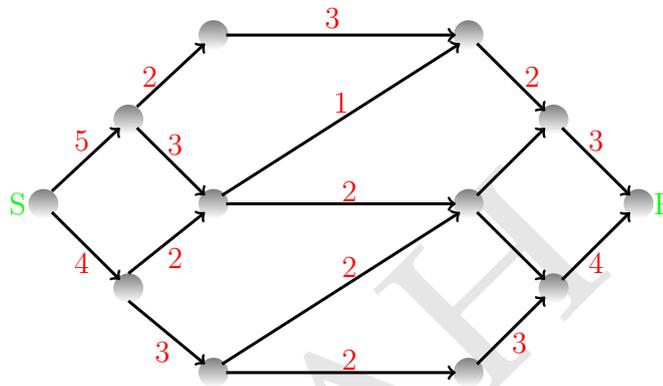


FIGURE 29 – Graphe $G_{52} = [X_{52}, U_{52}]$

La figure 29 représente un réseau dont S est le sommet source et P le sommet puits.

Exemple 32 (réseau) [Berge, 1985] Considérons le graphe $G_{53} = [X_{53}, U_{53}]$ de la figure 30 ci-dessous :

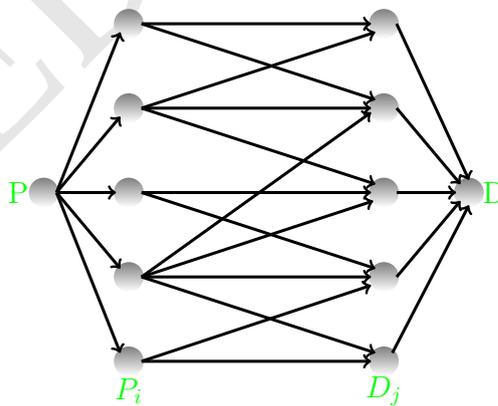


FIGURE 30 – Graphe $G_{53} = [X_{53}, U_{53}]$

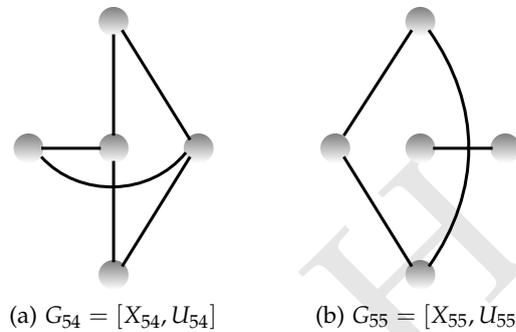
Dans la figure 30, les sommets P_i représentent des producteurs et les sommets D_j des distributeurs.

Lorsque $(P_i, D_j) \in U$, $c_{(P_i, D_j)}$ est le nombre maximum d'unités pouvant, durant une semaine, être acheminées par le producteur P_i au distributeur D_j . Ajoutons deux sommets P et D , et les arcs $(P, P_i), \forall i$, et $(D_j, D), \forall j$. De plus, posons $c_{(P, P_i)}$ = la production hebdomadaire maximum de P_i et $c_{(D_j, D)}$ = la distribution hebdomadaire maximum faite par D_j .

5.4 GRAPHE COMPLÉMENTAIRE

Définition 61 (graphe complémentaire) [Prins, 1994] Soit le graphe $G = [X, U]$. Le graphe complémentaire de G est le graphe $\bar{G} = [X, X^2 - U]$ dont les arcs (arêtes) sont tous(tes) les paires de sommets qui ne sont pas dans U .

Exemple 33 (graphe complémentaire) Les deux graphes de la figure 31 représentent respectivement, de gauche à droite, le graphe G_{54} et son complémentaire G_{55} .

FIGURE 31 – Graphe G_{54} et son complémentaire G_{55} [Hennecart et al., 2012]

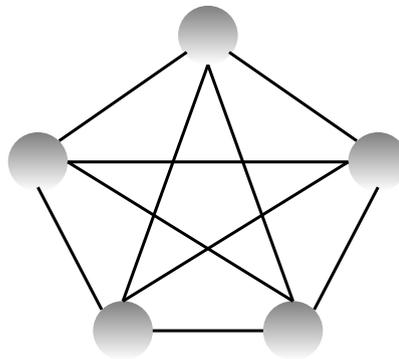
5.5 GRAPHE COMPLET

Définition 62 (graphe complet) [Prins, 1994] Un graphe est complet si toute paire de sommets est connectée par un arc ou une arête.

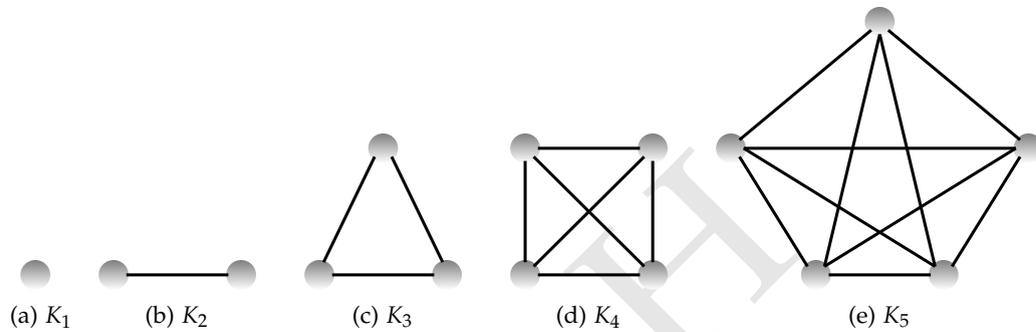
Définition 63 (graphe complet) Soit le graphe $G = [X, U]$. Le graphe G est dit complet si est seulement si :

- $\forall (x_i, x_j) \in X^2, x_i \neq x_j : (x_i, x_j) \in U$, si G est non orienté
- $\forall (x_i, x_j) \in X^2, x_i \neq x_j : (x_i, x_j) \in U \vee (x_j, x_i) \in U$, si G est orienté

Exemple 34 (graphe complet) Considérons le graphe G_{56} de la figure 32 ci-dessous :

FIGURE 32 – Graphe complet $G_{56} = [X_{56}, U_{56}]$

En l'honneur du topologiste polonais *Kuratowski*, un graphe **simple complet** d'ordre N est noté K_N (voir la figure 33).

FIGURE 33 – Graphes simples complets K_1 , K_2 , K_3 , K_4 et K_5

5.5.1 Stable

Définition 64 (stable) [Bondy et Murty, 2007] Un stable dans un graphe $G = [X, U]$ est l'ensemble de sommets dans X , deux à deux non-adjacents.

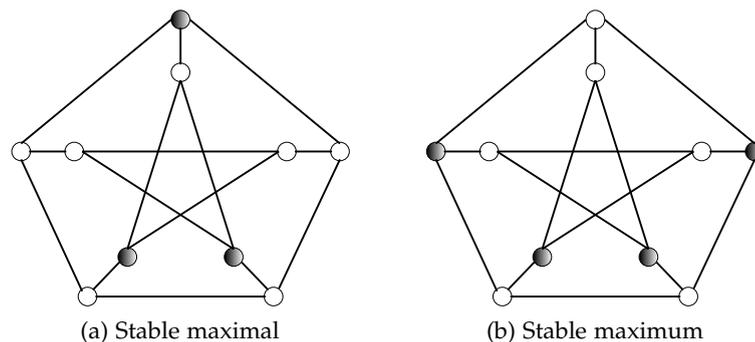
En d'autres termes, on peut définir un stable de $G = [X, U]$ par un ensemble de sommets $E = \{e_1, e_2, \dots, e_k\}$ qui vérifie les deux conditions :

1. $E \subseteq X$
2. $\forall (e_i, e_j) \in E^2, (e_i, e_j) \notin U$

Définition 65 (stable maximum et stable maximal) [Bondy et Murty, 2007] Soit le graphe $G = [X, U]$.

- Un stable dans G est maximum si G ne contient aucun stable de taille plus grande.
- Un stable dans G est maximal s'il ne peut pas être étendu en un stable plus grand.
- Un stable maximum est nécessairement maximal, mais pas inversement.
- Le cardinal d'un stable maximum dans G est la stabilité de G . Il est noté $\alpha(G)$.

Exemple 35 (stable et graphe de Peterson) Soit le graphe de Peterson $G_P = [X_P, U_P]$ de la figure 34 ci-dessous :

FIGURE 34 – Graphe G_P de Peterson

Dans le graphe (a) le stable est marqué par les sommets en gras. Il est maximal et non-maximum.

L'ensemble des sommets marqués en gras dans le graphe (b) représente un stable maximum. Par conséquent, le cardinal de G_p : $\alpha_{G_p} = 4$.

5.5.2 Clique

Définition 66 (clique) [Prins, 1994] Soit le graphe $G = [X, U]$. Une clique de G est un sous-graphe complet de G .

Définition 67 (clique maximum) Soit le graphe $G = [X, U]$. Une clique de G est maximum si G ne contient aucune clique de taille plus grande.

La taille maximum d'une clique de G (ou la cliquicité de G) est notée $\omega(G)$.

Exemple 36 (clique maximum) Considérons le graphe $G_{57} = [X_{57}, U_{57}]$ de la figure 35 ci-dessous :

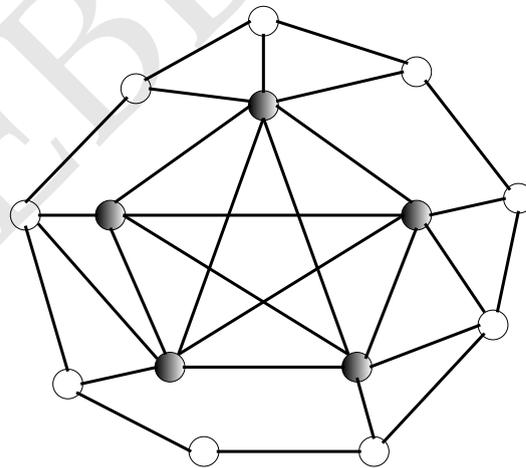


FIGURE 35 – Graphe simple $G_{57} = [X_{57}, U_{57}]$

L'ensemble des sommets du sous-graphe marqué en gras représente une clique maximum du graphe G_{57} . Donc, la cliquicité de G_{57} est $\omega(G) = 5$.

On peut noter qu'une clique S du graphe simple $G = [X, U]$ est un stable du graphe complémentaire \bar{G} de G . Ce qui implique la relation 10 ci-dessous :

$$\omega(G) = \alpha(\bar{G}) \quad (10)$$

Exemple 37 (clique maximum et stable maximum) Considérons les graphes simples donnés respectivement dans les figures 36a et 36b ci-dessous :



FIGURE 36 – Graphes G_{58} et \bar{G}_{58}

Nous avons :

- L'ensemble des sommets en gras de la figure 36a qui représente la clique maximum de G_{58} ,
- et l'ensemble des sommets en gras de la figure 36b qui donne le stable maximum de \bar{G}_{58} .

Nous constatons que la relation 10 est vérifiée : $\omega(G_{58}) = \alpha(\bar{G}_{58}) = 4$

5.6 GRAPHE SYMÉTRIQUE

Définition 68 (graphe symétrique) [Prins, 1994] Soit le graphe orienté $G = [X, U]$. Le graphe G est dit symétrique si l'existence de l'arc (x_i, x_j) implique celle de (x_j, x_i) .

En d'autres termes le graphe orienté $G = [X, U]$ est symétrique si l'implication suivante est vérifiée :

$$\forall (x_i, x_j) \in X^2 : (x_i, x_j) \in U \Rightarrow (x_j, x_i) \in U \tag{11}$$

Exemple 38 (graphe symétrique) Considérons le graphe $G_{59} = [X_{59}, U_{59}]$ de la figure 37 ci-dessous :

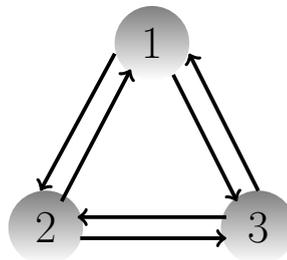


FIGURE 37 – Graphe $G_{59} = [X_{59}, U_{59}]$

Nous remarquons dans le graphe G_{59} , que :

- $(1, 2) \in U_{59} \wedge (2, 1) \in U_{59}$
- $(2, 3) \in U_{59} \wedge (3, 2) \in U_{59}$
- $(1, 3) \in U_{59} \wedge (3, 1) \in U_{59}$

Ce qui reflète la nature symétrique du graphe G_{59} .

Conclusion directe de la définition 68 et la relation 11 : « La matrice d'adjacence d'un graphe orienté symétrique doit être symétrique ».

5.7 GRAPHE ANTISYMMÉTRIQUE

Définition 69 (graphe antisymétrique) [Prins, 1994] Un graphe orienté $G = [X, U]$ est antisymétrique si l'existence de l'arc (x_i, x_j) implique la non existence de (x_j, x_i) .

En d'autres termes le graphe orienté $G = [X, U]$ est antisymétrique si l'implication suivante est vérifiée :

$$\forall (x_i, x_j) \in X^2 : (x_i, x_j) \in U \Rightarrow (x_j, x_i) \notin U \quad (12)$$

Exemple 39 (graphe antisymétrique) Considérons le graphe $G_{510} = [X_{510}, U_{510}]$ de la figure 38 ci-dessous :

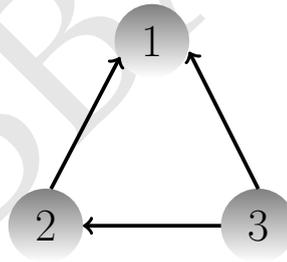


FIGURE 38 – Le graphe $G_{510} = [8_{510}, U_{510}]$

Nous remarquons dans le graphe G_{510} , que :

- $(2, 1) \in U_{510} \wedge (1, 2) \notin U_{510}$
- $(3, 2) \in U_{510} \wedge (2, 3) \notin U_{510}$
- $(3, 1) \in U_{510} \wedge (1, 3) \notin U_{510}$

Ce qui reflète la nature antisymétrique du graphe G_{510} .

Conclusion directe de la définition 69 et la relation 12 : « La matrice d'adjacence d'un graphe orienté antisymétrique n'est pas symétrique ».

5.8 GRAPHE BIPARTI

Définition 70 (graphe biparti) Un graphe $G = [X, U]$ est biparti si l'ensemble des sommet X admet une partition en deux sous-ensembles X_1 et X_2 , de telle sorte que chaque arête de U ait une extrémité dans X_1 et une autre dans X_2 . G est noté $G = [X_1, X_2, U]$.

Définition 71 (graphe biparti) Le graphe $G = [X, U]$ est dit biparti si et seulement si :

$$\begin{aligned} & \exists X_1, X_2 \in \mathcal{P}(X), X_1 \cap X_2 = \phi \wedge X_1 \cup X_2 = X : \\ & (x_i, x_j) \in U \Rightarrow (x_i, x_j) \in X_1 \times X_2 \vee (x_i, x_j) \in X_2 \times X_1 \end{aligned} \quad (13)$$

avec :

$\mathcal{P}(X)$: ensemble des partitions de X .

Définition 72 (graphe simple biparti complet) Un graphe simple biparti $G = [X_1, X_2, U]$ est complet si tout sommet de X_1 est relié à tous les sommets de X_2 .

Si $|X_1| = a$ et $|X_2| = b$ alors G est noté $K_{a,b}$.

Exemple 40 (graphe biparti) Considérons les graphes bipartis représentés par les figures 39a et 39b ci-dessous :

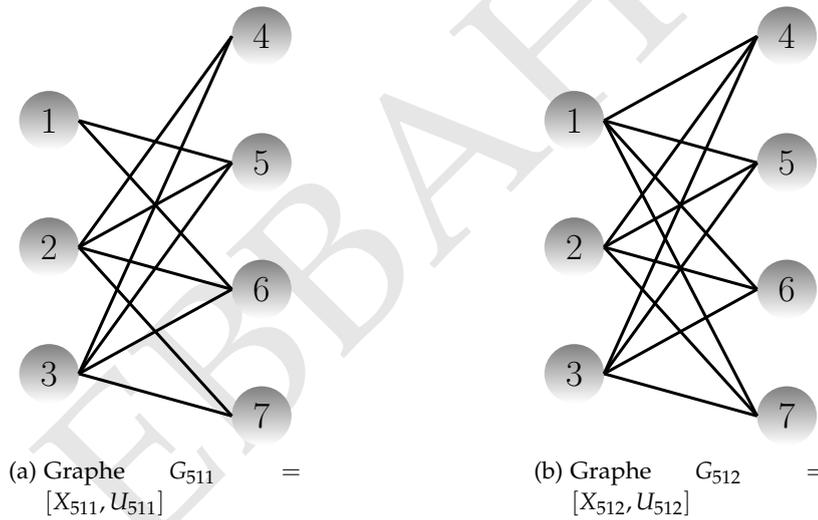


FIGURE 39 – Graphes bipartis

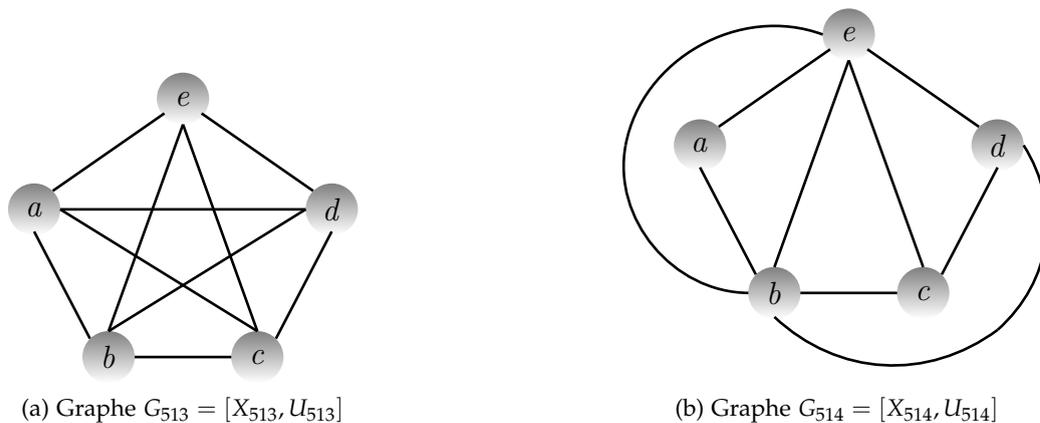
Le graphe G_{511} est biparti avec $X_{511_1} = \{1, 2, 3\}$ et $X_{511_2} = \{4, 5, 6, 7\}$.

Le graphe G_{512} est biparti complet avec $X_{512_1} = \{1, 2, 3\}$ et $X_{512_2} = \{4, 5, 6, 7\}$. Il est noté $K_{3,4}$ puisque $|X_{512_1}| = 3 \wedge |X_{512_2}| = 4$

5.9 GRAPHE PLANAIRE

Définition 73 (graphe planaire) Un graphe $G = [X, U]$ est dit planaire si on peut le dessiner dans un plan sans intersection d'arcs.

Exemple 41 (graphe planaire) Considérons les graphes représentés dans les figures 40a et 40b ci-dessous :



- G_{513} n'est pas planaire puisque nous ne pouvons pas le dessiner dans un plan sans avoir des arcs qui se croisent.
- G_{514} est planaire puisque nous pouvons le tracer dans un plan sans avoir des arcs qui se croisent.

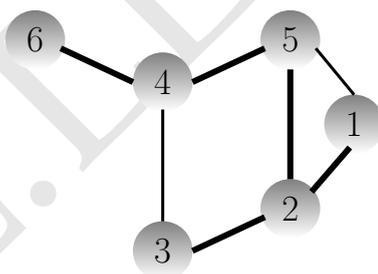
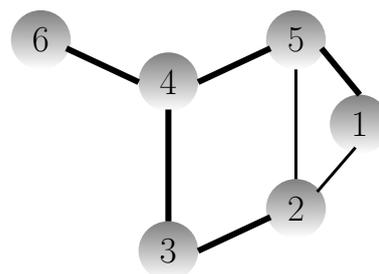
5.10 ARBRE

Définition 74 (arbre) [Gondran et Minoux, 1995] Un arbre est un graphe connexe sans cycles.

Propriété 1 [Gondran et Minoux, 1995] $\mathcal{I} = [X, T]$ est un arbre si et seulement si il existe une seule chaîne et une seule entre deux sommets quelconques.

Définition 75 (arbre d'un graphe) [Gondran et Minoux, 1995] Étant donné un graphe $G = [X, U]$, un arbre de G est un graphe partiel connexe et sans cycles de G .

Exemple 42 (arbres) Considérons les arbres du graphe G_{59} donnés à travers les figures 40c et 40d. Les arêtes qui forment l'arbre sont tracées en gras.

(c) Arbre \mathcal{I}_1 du graphe G_{59} (d) Arbre \mathcal{I}_2 du graphe G_{59} FIGURE 40 – Arbres dans le graphe $G_{59} = [X_{59}, U_{59}]$

Nous remarquons que :

- \mathcal{I}_1 est un arbre qui contient 4 arêtes,
- \mathcal{I}_2 est un arbre qui contient 5 arêtes.

\mathcal{I}_2 est un arbre maximal puisque nous ne pouvons pas obtenir un arbre plus grand dans G_{59} .

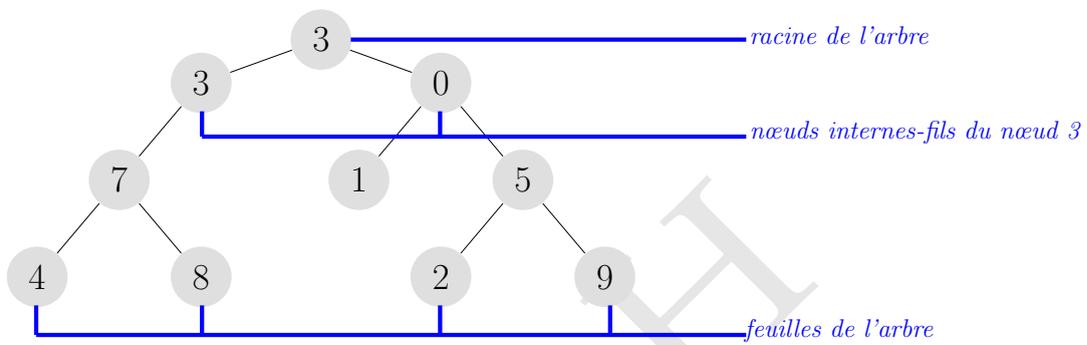
5.10.1 Terminologie

Les arbres possèdent une terminologie qui est conforme à sa structure arborescente. Ils contiennent trois types de nœuds (voir figure 41) :

racine : comme son nom l'indique, c'est le nœud qui représente la racine de l'arbre,

nœud interne : nœud qui a des nœuds descendants (ou des enfants),

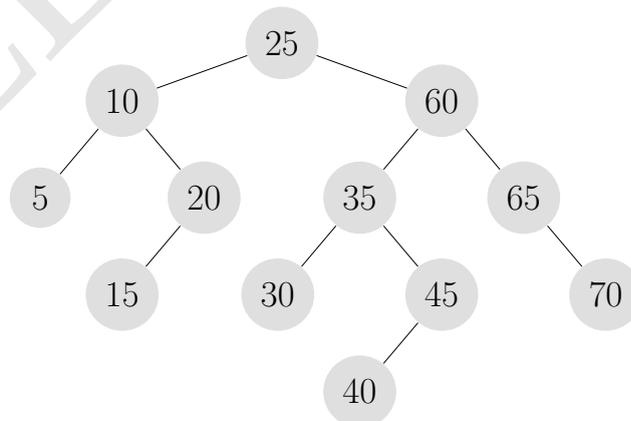
feuille : nœud qui n'a pas de nœuds descendants (ou des enfants).

FIGURE 41 – Arbre $\mathcal{T}' = [X', T']$

5.10.2 Arbre binaire

Définition 76 (arbre binaire) Un arbre binaire est un arbre dont le nœud ne peut pas avoir plus de deux fils, appelés respectivement fils gauche et fils droit.

Exemple 43 (arbre binaire de recherche) Considérons l'arbre binaire de recherche \mathcal{T}'' de la figure 42 ci-dessous :

FIGURE 42 – Arbre $\mathcal{T}'' = [X'', T'']$

Dans un arbre de recherche tel \mathcal{T}'' , nous remarquons que la valeur du fils gauche est plus petite que celle du père, et la valeur du fils droit plus grande. Par conséquent, la valeur de la racine doit être :

- plus grande que les valeurs contenues dans le sous-arbre gauche,
- et plus petite que les valeurs contenues dans le sous-arbre droit.

ALGORITHMES DE BASE EN THÉORIE DES GRAPHS

INTRODUCTION

Le plus court chemin et le flot maximum sont des problèmes qui peuvent être projetés sur d'autres issus de l'industrie, de l'économie, d'un domaine technique (informatique, automatique, électricité, etc.), etc.

Dans ce chapitre nous présentons deux types d'algorithmes du plus court chemin et un algorithme du flot maximum.

Nous introduisons chacun des algorithmes avec la description du problème et un exemple illustratif.

6.1 ALGORITHMES DU PLUS COURT CHEMIN

Les algorithmes de résolution diffèrent suivant les propriétés du graphe $G = [X, U]$ d'ordre N [Gondran et Minoux, 1995] :

- $l(u) \geq 0, \forall u \in U,$
- $l(u) = 0, \forall u \in U,$
- G et $l(u)$ quelconques,
- G sans circuit.

et suivant le problème à traiter :

- recherche du plus court chemin d'un sommet à un autre,
- recherche du plus court chemin d'un sommet à tous les autres,
- recherche du plus court chemin entre toute paire de sommets.

Parmi les algorithmes de résolution du problème du plus court chemin nous pouvons citer :

- Algorithme de Berge-1958 : recherche du plus court chemin d'un sommet à tous les autres en utilisant la liste de succession.
- Algorithme de Bellman-1958 : recherche du plus court chemin d'un sommet à tous les autres dans un graphe quelconque (ou détection d'un circuit négatif).
- Algorithme de Ford-1956 : recherche du plus court chemin d'un sommet à tous les autres dans un graphe quelconque en utilisant la fonction multi-voque inverse Γ^{-1} .
- Algorithme de Floyd-1962 : recherche de la matrice des plus courts chemins pour un graphes avec $l(u)$ quelconque.

- Algorithme de Dantzing-1966 : recherche de la matrice des plus courts chemins et détection du cycle de longueur négative.
- Algorithme de Floyd-Warshall : recherche de la matrice des plus courts chemins pour un graphe sans circuit de longueur négative.

Dans le présent cours nous présentons uniquement deux algorithmes :

1. algorithme de recherche du plus court chemin d'un sommet à tous les autres, avec G quelconque et $l(u) \geq 0, \forall u \in U$,
2. algorithme de recherche du plus court chemin d'un sommet à tous les autres, avec G et $l(u)$ quelconques.

Dans ces deux algorithmes nous posons :

- $X = \{1, 2, \dots, N\}$,
- l_{ij} la longueur de l'arc $(i, j) \in U$,
- $\pi^*(i)$ la longueur minimum des chemins de 1 à i ; en particulier $\pi^*(1) = 0$, avec 1 sommet de départ des chemins à calculer.

6.1.1 Graphe à longueurs positives

L'algorithme MOOREDIJKSTRA 1959 1 procède en $N - 1$ itérations. Au début de chaque itération l'ensemble des sommets est partitionné en deux sous-ensembles S et $\bar{S} = X - S$, avec $1 \in S$.

Chaque sommet est assigné d'une étiquette $\pi(i)$ telle que :

- si $i \in S, \pi(i) = \pi^*(i)$
- si $i \in \bar{S}, \pi(i) = \min_{\substack{k \in S \\ k \in \Gamma_i^{-1}}} (\pi(k) + l_{ki})$

Par conséquent, la valeur $\pi(i) (i \in \bar{S})$ donne la longueur minimale des chemins de 1 à i , soumis à la condition que tous ses sommets excepté i sont dans S .

La pertinence de l'algorithme dépend du lemme suivant :

lemme 1 [Gondran et Minoux, 1995]

Si $i \in \bar{S}$ tel que $\pi(j) = \min_{i \in \bar{S}} \pi(i)$ alors $\pi^*(j) = \pi(j)$.

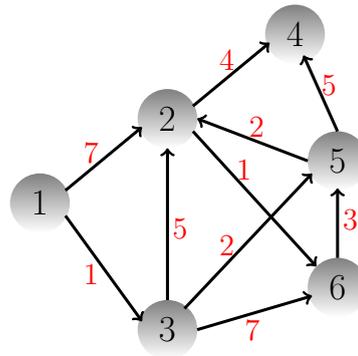
Algorithm 1 MOOREDIKJSTRA 1959**Input:** $G = [X, U]$ le graphe à traiter**Output:** Recherche du plus court chemin du sommet 1 aux autres dans un graphe de longueurs positives

```

1: Initialisation :  $\bar{S} = \{2, 3, \dots, N\}$ ,  $\pi(1) = 0$   $\pi(i) = \begin{cases} l_{1i} & \text{si } i \in \Gamma_1 \\ +\infty & \text{sinon} \end{cases}$ 
2: Sélectionner  $j \in \bar{S}$  tel que  $\pi(j) = \min_{i \in \bar{S}} \pi(i)$ 
3:  $\bar{S} = \bar{S} - \{j\}$ 
4: if  $|\bar{S}| = 0$  then
5:   FIN (plus court chemin calculé)
6: else aller en (8)
7: end if
8: for  $i \in \Gamma_j$  et  $i \in \bar{S}$  do
9:    $\pi(i) \leftarrow \min(\pi(i), \pi(j) + l_{ji})$ 
10: end for
11: retourner en (2)

```

Exemple 44 (application de l'algorithme 1) Soit le graphe orienté pondéré G_{61} de la figure ci-dessous 43 :

FIGURE 43 – $G_{61} = [X_{61}, U_{61}]$ [Gondran et Minoux, 1995]

Nous cherchons les plus courts chemins allant du sommet **1** à tous les autres.

Initialement, nous avons :

$$\bar{S} = \{2, 3, 4, 5, 6\},$$

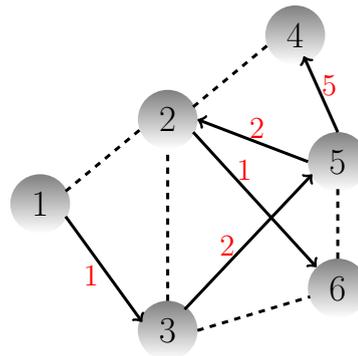
$$\text{et } \pi(1) = 0, \pi(2) = 7, \pi(3) = 1, \pi(4) = \pi(5) = \pi(6) = +\infty$$

Nous récapitulons les étapes d'application de l'algorithme sur le graphe à travers le tableau ci-dessous 3 :

TABLE 3 – Table de déroulement de l'algorithme 1 sur le graphe G_{61}

j	\bar{S}	$\Gamma_j \cap \bar{S}$	$\pi(i)$ (i :successeur de j)
3	$\{2, 4, 5, 6\}$	$\{2, 5, 6\}$	$\pi(2) = \min(7, 1 + 5) = 6$
			$\pi(5) = \min(+\infty, 1 + 2) = 3$
			$\pi(6) = \min(+\infty, 1 + 7) = 8$
5	$\{2, 4, 6\}$	$\{2, 4\}$	$\pi(2) = \min(6, 3 + 2) = 5$
			$\pi(4) = \min(+\infty, 3 + 5) = 8$
2	$\{4, 6\}$	$\{4, 6\}$	$\pi(4) = \min(8, 5 + 4) = 8$
			$\pi(6) = \min(8, 5 + 1) = 6$
6	$\{4\}$	\emptyset	/
4	\emptyset	\emptyset	/

La dernière ligne du tableau indique la fin de l'algorithme puisque $\bar{S} = \emptyset$.
Les plus courts chemins tracés par l'algorithme se résume via le graphe illustré dans la figure 44.

FIGURE 44 – Plus courts chemins obtenus sur G_{61} (figure 43)[Gondran et Minoux, 1995]

6.1.2 Graphe à longueurs quelconques

L'algorithme 2 MOORE 1957 converge en N itérations. Il est applicable dans le cas de recherche du plus court chemin d'un sommet à tous les autres sur un graphe quelconque.

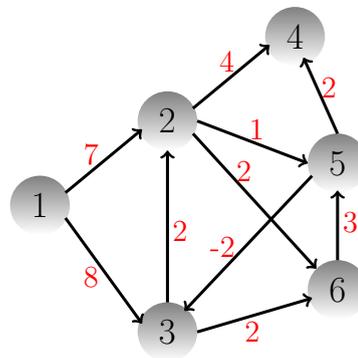
Algorithm 2 MOORE 1957**Input:** $G = [X, U]$ le graphe à traiter**Output:** Recherche du plus court chemin du sommet 1 aux autres dans un graphe dont les longueurs sont quelconques

```

1: Initialisation :  $\bar{S} = \{2, 3, \dots, N\}$ ,  $\pi(1) = 0$   $\pi(i) = \begin{cases} l_{1i} & \text{si } i \in \Gamma_1 \\ +\infty & \text{sinon} \end{cases} \quad i \geq 2$ 
2: Sélectionner  $j \in \bar{S}$  tel que  $\pi(j) = \min_{i \in \bar{S}} \pi(i)$ 
3:  $\bar{S} = \bar{S} - \{j\}$ 
4: for  $i \in \Gamma_j$  do
5:   calculer  $\pi^* = \pi(j) + l_{ji}$ 
6:   if  $\pi^* < \pi(i)$  then
7:      $\bar{S} \leftarrow \bar{S} + \{i\}$  et  $\pi(i) = \pi^*$ 
8:   end if
9: end for
10: if  $|\bar{S}| = 0$  then
11:   FIN
12: else
13:   aller en (2)
14: end if

```

Exemple 45 (application de l'algorithme 2) Soit le graphe orienté valué à longueurs quelconques G_{62} de la figure 45.

FIGURE 45 – $G_{62} = [X_{62}, U_{62}]$ [Gondran et Minoux, 1995]

Nous cherchons les plus courts chemins allant du sommet 1 à tous les autres.

Initialement, nous avons :

$$\bar{S} = \{2, 3, 4, 5, 6\},$$

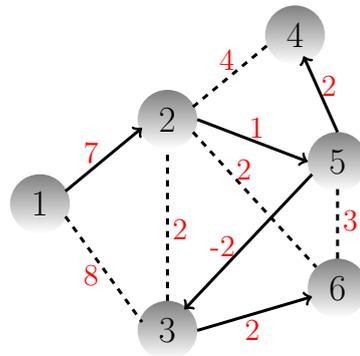
$$\text{et } \pi(1) = 0, \pi(2) = 7, \pi(3) = 8, \pi(4) = \pi(5) = \pi(6) = +\infty$$

Nous récapitulons les étapes d'application de l'algorithme sur G_{62} à travers le tableau 4 :

TABLE 4 – Table de déroulement de l’algorithme 2 sur le graphe G_{61}

j	\bar{S}	Γ_j	$\pi(i)$ (i :successeur de j)
2	$\{3, 4, 5, 6\}$	$\{4, 5, 6\}$	$\pi(4) = \min(+\infty, 7 + 4) = 11$
			$\pi(5) = \min(+\infty, 7 + 1) = 8$
			$\pi(6) = \min(+\infty, 7 + 2) = 9$
3	$\{4, 5, 6\}$	$\{2, 6\}$	$\pi(2) = \min(7, 8 + 2) = 7$
5	$\{4, 6\}$ $\{3, 4, 6\}$	$\{4, 3\}$	$\pi(4) = \min(9, 8 + 2) = 9$
			$\pi(3) = \min(8, 8 + (-2)) = 6$
3	$\{4, 6\}$	$\{2, 6\}$	$\pi(2) = \min(7, 6 + 2) = 7$
6	$\{4\}$	$\{5\}$	$\pi(5) = \min(8, 8 + 3) = 8$
4	\emptyset	\emptyset	/

La dernière ligne du tableau indique la fin de l’algorithme puisque $\bar{S} = \emptyset$
 Les plus courts chemins tracés par l’algorithme se résume via le graphe illustré dans la figure 46.

FIGURE 46 – Plus courts chemins obtenus sur G_{62} (figure 45)

6.2 ALGORITHME DU FLOT MAXIMUM

La recherche d’un flot maximum dans un réseau est un problème perçu dans différents domaines, à savoir : le transport (routier, aérien, ferroviaire), la structuration et le dimensionnement optimaux des réseaux de communication, la gestion des stocks, l’ordonnancement, sans oublier son intérêt considérable dans les mathématique combinatoires [Gondran et Minoux, 1995].

L’un des meilleurs exemples illustratifs de la notion du flot sur un graphe est certainement celui du courant électrique continu parcourant un réseau de dipôles.

Dans cette section nous faisons appel aux notions vues dans la section 5.3 du chapitre 5.

Définition 77 (réseau de transport) Un *réseau de transport* est un graphe orienté $G = [X, U]$ avec :

- une capacité $\gamma(u) \geq 0$ assignée à chaque arc $u \in U$,
- deux sommets particuliers $s, p \in X$ appelés respectivement, source et puits de G .

Définition 78 (flux d'un arc) Soient le réseau $G = [X, U]$ et le flot φ dans G . Le flux d'un arc $u \in U$ est la quantité $\varphi(u)$ porté par u .

Définition 79 (flot) Soit le réseau $G = [X, U]$ dont les arcs sont numérotés de 1 à M . Un flot dans G est une application :

$$\begin{aligned} \varphi : U &\longrightarrow \mathbb{R}^M \\ u &\longmapsto (\varphi_{u_1}, \varphi_{u_2}, \dots, \varphi_{u_M}) \end{aligned}$$

qui vérifie les conditions :

$$\forall u \in U : 0 \leq \varphi_u \leq \gamma(u) \quad (14)$$

et

$$\forall x \in X - \{s, p\} : \sum_{u \in \omega^-(\{x\})} \varphi_u = \sum_{u \in \omega^+(\{x\})} \varphi_u \quad (15)$$

La relation 15 exprime simplement que la somme des flux entrants en un sommet, est égale à la somme des flux sortants. Elle exprime la loi des nœuds de **Kirchhoff**.

On considère le graphe $G^0 = [X, U^0]$ déduit de G en rajoutant l'arc (p, s) . Ce dernier, appelé **arc de retour** du flot, est assigné par le numéro 0. En d'autres termes, les arcs de G^0 sont numérotés de 0 à M .

La capacité φ_0 est appelée la **valeur du flot**.

Si $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_M)^T$ est un flot de s à p dans G de valeur φ_0 , alors $\varphi' = (\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_M)^T$ est simplement un flot dans G^0 avec :

$$\sum_{u \in \omega^+(\{s\})} \varphi_u = \sum_{u \in \omega^-(\{p\})} \varphi_u = \varphi_0 \quad (16)$$

Définition 80 (graphe d'écart) Soit le flot $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_M)^T$ entre s et p dans $G [X, U]$.

Le graphe d'écart associé à φ est le graphe $\bar{G}(\varphi) = [X, \bar{U}(\varphi)]$ ayant le même ensemble de sommets X et dont chaque arc $u \in \bar{U}(\varphi)$ est constitué de la manière suivante :

- à chaque arc $u = (i, j) \in U$ de G , on associe au plus deux arcs de $\bar{U}(\varphi)$

$$u^+ = (i, j) \text{ si } \varphi_u < \gamma(u) \text{ et } u^- = (j, i) \text{ si } \varphi_u > 0.$$
- on associe à chaque arc de $\bar{G}(\varphi)$ une capacité résiduelle égale à :
 - $\gamma(u) - \varphi_u > 0$ pour u^+ ;
 - $\varphi_u > 0$ pour u^- .

Théorème 3 (flot maximum) Soit φ un flot de s à p dans $G = [X, U]$.

L'absence d'un chemin de s à p dans $\bar{G}(\varphi)$ est une condition nécessaire et suffisante pour que φ soit maximal.

L'algorithme de *FORD-FULKERSON* de résolution du problème du flot maximum se donne comme suit :

Algorithm 3 Algorithme de FORD-FULKERSON

Input: $G = [X, U]$ le graphe à traiter ;

Output: Recherche du flot maximal

$k = 0$

Choisir un flot initial φ^0 compatible avec les contraintes de capacité. Par exemple :

$\varphi^0 = (0, 0, \dots, 0)$

$It = true$

while It **do**

Rechercher un chemin π^k de s à p dans le graphe d'écart $\bar{G}(\varphi^k)$

if π^k n'existe pas **then**

φ^k est un flot maximum

$It = false$

else

Soit ϵ^k la capacité résiduelle de π^k (minimum des capacités résiduelles des arcs du chemin)

Définir le flot φ^{k+1} par :

$$\varphi_u^{k+1} = \varphi_u^k + \epsilon^k \text{ si } u^+ \in \pi^k$$

$$\varphi_u^{k+1} = \varphi_u^k - \epsilon^k \text{ si } u^- \in \pi^k$$

$$\varphi_0^{k+1} = \varphi_0^k + \epsilon^k$$

$k = k + 1$

end if

end while

return φ^k

Exemple 46 (application de l'algorithme 3) Soit le graphe orienté valué G_{63} de la figure ci-dessous 47 :

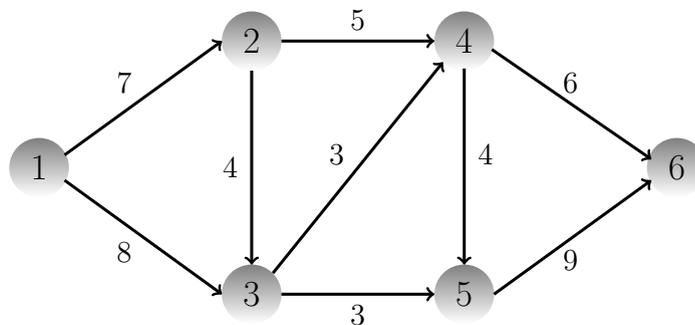
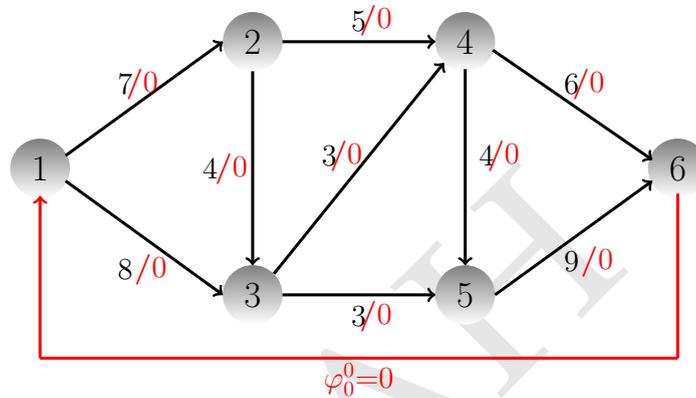


FIGURE 47 - $G_{63} = [X_{63}, U_{63}]$

Nous cherchons un flot maximum du graphe 47 avec application de l'algorithme 3.

Tel illustré dans la figure 48, nous avons un flot initial nul $\varphi_0^0 = 0$ avec $k = 0$ et $It = true$.

FIGURE 48 – Flot nul du graphe G_{63}

Nous récapitulons les étapes d'application de l'algorithme sur le graphe de la figure 47 à travers le tableau 5.

Chaque ligne du tableau correspond à l'itération k de l'algorithme. Nous donnons le chemin choisi π_k^1 et le graphe d'écart correspondant au flot φ^k .

La troisième colonne illustre, dans l'ordre des itérations, les chemins et les flots à travers les figures 49, 51 et 53.

Les graphes d'écart correspondants aux flots φ^1 , φ^2 et φ^3 sont donnés respectivement, via les figures 50, 52 et 54 de la quatrième colonne.

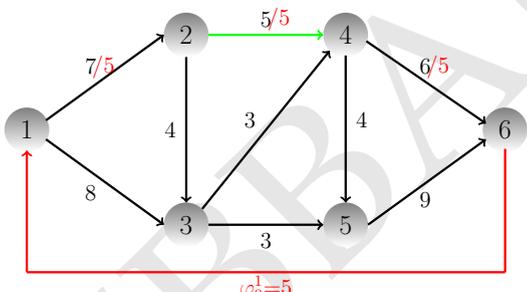
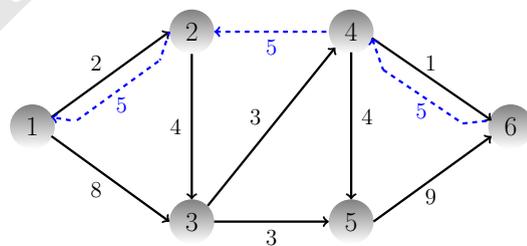
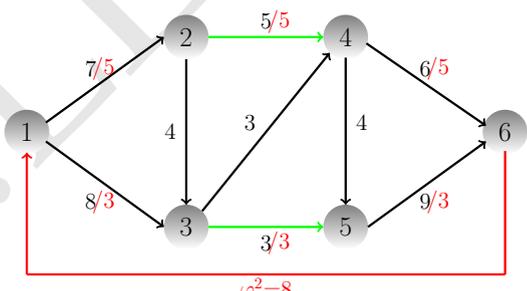
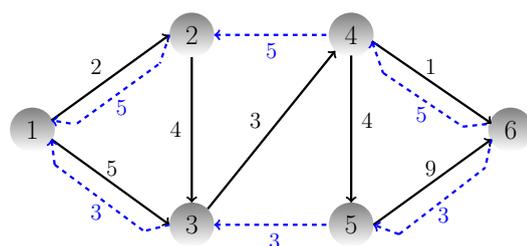
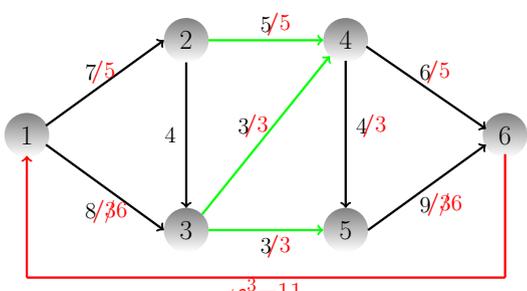
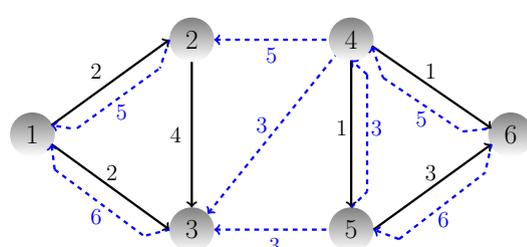
La dernière ligne du tableau indique la fin de l'algorithme puisque $It = false$. En d'autres termes aucune amélioration du flot φ^3 de valeur 11.

Le flot maximum n'est pas unique, mais sa valeur est unique. En d'autres termes, on peut trouver d'autres flots maximums, mais de valeur max inchangée 11.

1. Le chemin π_k n'est pas unique. Il faut choisir le chemin qui me permet d'avoir plus d'arc saturés dans le flot.

2. La graphe d'écart dépend du chemin choisi π_k

TABLE 5 – Table de déroulement de l’algorithme 3 sur le graphe G_{63}

It	k	π_k et φ^{k+1}	graphe d'écart
<i>true</i>	0	 <p>FIGURE 49 – Flot φ^1 de G_{63}</p>	 <p>FIGURE 50 – $\bar{G}(\varphi^1)$</p>
<i>true</i>	1	 <p>FIGURE 51 – Flot φ^2 de G_{63}</p>	 <p>FIGURE 52 – $\bar{G}(\varphi^2)$</p>
<i>true</i>	2	 <p>FIGURE 53 – Flot φ^3 de G_{63}</p>	 <p>FIGURE 54 – $\bar{G}(\varphi^3)$</p>
<i>true</i>	3	pas de chemin π_3	/
<i>false</i>	/	/	/

Deuxième partie

INTRODUCTION À LA PROGRAMMATION LINÉAIRE

INTRODUCTION

La Programmation Linéaire (PL) est une branche de la Programmation Mathématique conçue pour la résolution des problèmes d'optimisation combinatoire. Ces problèmes peuvent être issus de domaines divers de l'industrie, à savoir :

- les finances,
- la planification et le contrôle d'une chaîne de production,
- l'énergie telle que le pétrole, le gaz, l'électricité et le nucléaire,
- le transport, que ce soit aériens, routiers ou ferroviaires,
- télécommunications,
- industrie forestière,
- etc.

La PL représente un outil puissant pour la résolution des problèmes modélisables sous forme d'un programme linéaire (programme avec des contraintes linéaires). En 1947, *George Dantzing* a développé l'algorithme *SIMPLEXE*, une méthode qui s'est imposée par son efficacité sur les programmes linéaires. Dans une enquête réalisée sur *Fortune 500*³, 85% des entreprises déclarent avoir utilisé la programmation linéaire [Winston, 1991].

Dans cette partie, nous mettons en avant deux étapes fondamentales de la PL : (1) modélisation d'un problème sous un PL, et (2) résolution du PL via l'application du *SIMPLEXE*. Elle se résume en deux chapitres :

- le premier chapitre introduit les notions de base de la programmation linéaire et la modélisation d'un problème d'optimisation en un PL. A ce niveau nous exposons la technique de résolution graphique (dans le cas d'un problème à deux variables) et l'algorithme du *SIMPLEXE* avec des exemples d'application.
- le deuxième chapitre décrit le principe de la dualité est la résolution du problème dual.

3. *Fortune 500* est le classement des 500 premières entreprises américaines, classées selon l'importance de leur chiffre d'affaires. Réalisée initialement en 1955, cette liste prestigieuse est publiée chaque année par le magazine *Fortune*.

RÉSOLUTION D'UN PROGRAMME LINÉAIRE

7.1 INTRODUCTION

La programmation linéaire constitue un problème fondamental de l'optimisation combinatoire. Elle est conçue pour appréhender les problèmes linéaires ou plus précisément les problèmes modélisables en contraintes linéaires.

Dans l'exemple 47 ci-dessous nous illustrons :

- le passage d'un énoncé décrivant le problème à résoudre vers le programme linéaire,
- puis la résolution graphique du problème et le calcul de la solution optimale

Exemple 47 (solution graphique d'un PL à 2 variables [Charon et al., 1996]) Une usine fabrique 2 sortes de produits p_1 et p_2 à l'aide de 2 machines m_1 et m_2 . Chaque unité de produit encourt de fabrication doit passer successivement sur les 2 machines dans un ordre indifférent et pendant les temps suivants (en minutes) :

	p_1	p_2
m_1	30	20
m_2	40	10

De plus la machine m_1 est disponible 6000min / mois et la machine m_2 est disponible 4000min / mois.

Le profit réalisé sur une unité du produit p_1 est de 400u.m.

Le produit réalisé sur une unité du produit p_2 est de 200u.m.

On souhaite trouver le plan de fabrication mensuel qui maximise le profit.

Pour cela appelons :

- x : le nombre d'unités du produit p_1 à fabriquer mensuellement,
- y : le nombre d'unités du produit p_2 à fabriquer mensuellement.

On voit que ce problème peut s'exprimer sous la forme suivante :

$$\begin{aligned} \max z &= 400x + 200y \\ \text{s.c} \left\{ \begin{array}{l} 30x + 20y \leq 6000 \\ 40x + 10y \leq 4000 \\ x \geq 0, y \geq 0 \end{array} \right. & \quad (P_1) \end{aligned} \quad (17)$$

Le problème étant en 2 variables, il admet une solution graphique facile à mettre en œuvre.

Nous commençons par représenter les contraintes graphiquement. Les points (x, y) qui satisfont les contraintes appartiennent au quadrilatère en gris ABCO, tel qu'illustré dans la figure 55.

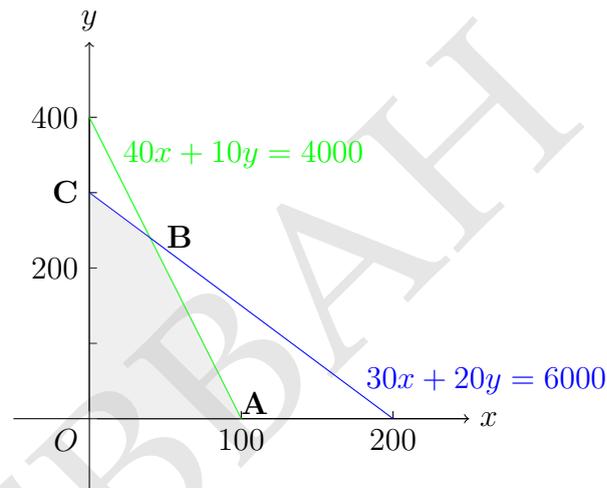


FIGURE 55 – Représentation des contraintes de (P_1) (voir le modèle 17)

La famille des droites $D_\lambda = \{(x, y) / 400x + 200y = \lambda\}$ est une famille de droites parallèles.

Pour représenter la fonction objectif, on fixe λ à 40000 ce qui donne la droite en gras qui passe par l'espace des solutions réalisables tel qu'il est illustré dans la figure 56.

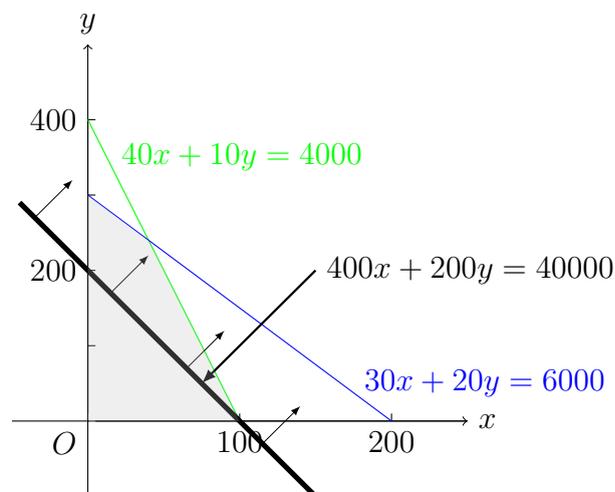
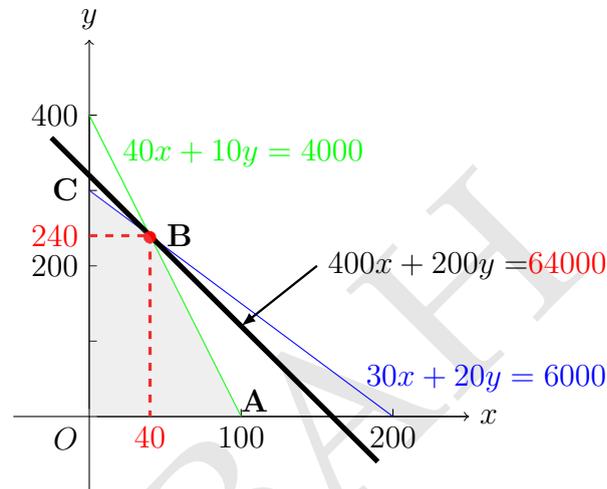


FIGURE 56 – Représentation de la fonction objectif de (P_1) (voir le modèle 17)

En faisant déplacer la droite $400x + 200y = \lambda$ vers le haut, on s'approche de plus en plus de la solution optimale (voir la figure 57).

FIGURE 57 – Résolution graphique de (P_1) (voir le modèle 17)

Parmi les droites D_λ qui ont une intersection non-vide avec le quadrilatère est celle qui passe par le point B. Elle correspond à la plus grande valeur $\lambda = 64000$ et elle passe par un seul point du quadrilatère $B = (40, 240)$.

Par conséquent, la solution optimale à notre problème est donc $(x^*, y^*) = (40, 240)$ avec $z^* = 64000$.

7.2 STRUCTURE D'UN PL

Un problème de programmation linéaire est un problème qui peut s'exprimer sous forme d'un programme linéaire (PL).

La structure d'un PL peut se définir via :

1. **les variables de décision** : n variables de décision (ou de choix) : $x_j \geq 0, j = 1, \dots, n$
2. **la fonction objectif** : une fonction de maximisation (par exemple : le profit ou le taux de production) ou de minimisation (par exemple : le coût de la production ou le temps de fabrication) en fonction des n variables x_1, x_2, \dots, x_n . Elle reflète l'aspect optimisation du problème,
3. **les m contraintes linéaires** : sont des équations ou des inéquations linéaires sous la forme suivante :

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, i = 1, \dots, m,$$

Donc, un (PL) sous la forme standard se donne comme suit :

$$\begin{aligned} \max z &= \sum_{j=1}^n c_j x_j \\ \text{s.c} \left\{ \begin{array}{l} \sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, \dots, m \\ x_j \geq 0, j = 1, \dots, n \end{array} \right. & \quad (PL_{st}) \end{aligned} \quad (18)$$

7.3 ALGORITHME DU SIMPLEXE

L'algorithme du SIMPLEXE a été conçu en 1947 par G.B. Dantzig. Il est dédié à la résolution des programmes linéaires à variables continues.

Considérons un (PL) écrit sous la forme standard donnée par le modèle 18. Nous exprimons le SIMPLEXE via l'algorithme 4 qui se résume en huit étapes :

Algorithm 4 SIMPLEXE

Input: Un PL de la forme 18

Output: Recherche de la solution optimale du PL

1: Intégrer les variables d'écart :

$$\begin{aligned} x_{n+i} &= b - \sum_{j=1}^n a_{ij}x_j, i = 1, \dots, m \\ z &= \sum_{j=1}^n c_jx_j \end{aligned} \quad \text{Dictionnaire I}$$

2: Calculer la première solution réalisable :

$$(x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}) = (0, \dots, 0, b_1, b_2, \dots, b_m)$$

3: **if** $\exists c_i > 0 / i = 1..n$ **then**

4: Calculer la variable entrante x_r :

$$x_r, r \in 1..n / c_r = \max_{i=1..n}(c_i)$$

5: **else**

6: Aller en 11

7: **end if**

8: Calculer la variable sortante x_s qui est la première à s'annuler telle que :

$$\frac{b_s}{a_{sr}} = \min_{i=1..m} \left(\frac{b_i}{a_{ir}} \right) / a_{ir} > 0$$

9: Exprimer le nouveau dictionnaire, remplacer :

$$\begin{aligned} &\text{- les } a_{ij} \text{ par } a_{ij} - a_{jr} \frac{a_{si}}{a_{sr}} \\ &\text{- les } b_j \text{ par } b_j - a_{jr} \frac{b_s}{a_{sr}} \end{aligned}$$

10: Retourner en 3

11: FIN(La solution est optimale).

Exemple 48 (application du Simplexe[Charon et al., 1996]) Une fabrique d'objets en terre cuite peinte produit des cendriers, des cruches, des bols et des vases. La fabrication de chacun de ces objets nécessite un certain nombre d'heures de moulage, un certain nombre d'heures de cuisson et un certain nombre d'heures de peinture. La vente de ces objets rapporte un certain bénéfice. Ces données sont récapitulées dans le tableau suivant :

Objet	Cendrier	Bol	Cruche	Vase
Moulage	2	4	5	7
Cuisson	1	1	2	2
Peinture	1	2	3	3
Bénéfice	7	9	18	17

L'entreprise dispose, quotidiennement, de 42 heures de moulage, 17 heures de cuisson et 24 heures de peinture.

On souhaite établir un plan de fabrication de façon à maximiser le chiffre d'affaires (on suppose que l'on est en régime stable de fabrication et non en phase initiale où il faut mouler avant de cuire et cuire avant de peindre).

Appelons :

x_1 : nombre de cendriers,

x_2 : nombre de cruches,

x_3 : nombre de bols,

x_4 : nombre de vases.

Le problème admet le programme linéaire (P_2) (voir le modèle 19) comme suit :

$$\begin{aligned} \max z &= 7x_1 + 9x_2 + 18x_3 + 17x_4 \\ \text{s.c} \begin{cases} 2x_1 + 4x_2 + 5x_3 + 7x_4 & \leq 42 \\ x_1 + x_2 + 2x_3 + 2x_4 & \leq 17 \\ x_1 + 2x_2 + 3x_3 + 3x_4 & \leq 24 \\ x_1, x_2, x_3, x_4 & \geq 0 \end{cases} \end{aligned} \quad (P_2) \quad (19)$$

Initialement, on écrit (P_2) sous la forme canonique ($PLst_2st$) en introduisant trois variables d'écart x_5, x_6 et x_7 , ce qui donne :

$$\begin{aligned} \max z &= 7x_1 + 9x_2 + 18x_3 + 17x_4 \\ \text{s.c} \begin{cases} 2x_1 + 4x_2 + 5x_3 + 7x_4 + x_5 & = 42 \\ x_1 + x_2 + 2x_3 + 2x_4 + x_6 & = 17 \\ x_1 + 2x_2 + 3x_3 + 3x_4 + x_7 & = 24 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 & \geq 0 \end{cases} \end{aligned} \quad (P_{2st}) \quad (20)$$

Ce qui s'exprime par le premier dictionnaire D_1 , instruction 1 de l'algorithme 4. Il se donne comme suit :

$$\begin{array}{r} x_5 = 42 - 2x_1 + 4x_2 + 5x_3 + 7x_4 \\ x_6 = 17 - x_1 + x_2 + 2x_3 + 2x_4 \\ x_7 = 24 - x_1 + 2x_2 + 3x_3 + 3x_4 \\ \hline z = 7x_1 + 9x_2 + 18x_3 + 17x_4 \end{array} \quad D_1$$

Le problème s'écrit maintenant avec les contraintes $x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0$.

Le polyèdre des contraintes est limité par les hyperplans d'équation $x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 0, x_7 = 0$.

On distingue deux types de variables :

- variables de base = $\{x_5, x_6, x_7\}$,
- variables hors-base = $\{x_1, x_2, x_3, x_4\}$.

La recherche de la solution optimale du D_1 , se fait en suivant les étapes suivantes :

1. recherche de la solution basique (instruction 2 de l'algorithme 4) : elle est obtenue en donnant la valeur 0 aux variables hors-base. Par conséquent, la solution initiale (ou basique) sera : $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*) = (0, 0, 0, 0, 42, 17, 24)^T$ et $z = 0$. $x_i^* \geq 0 / i \in 1..7$ ce qui rend le dictionnaire réalisable.

1. x^* représente la valeur prise par la variable x

2. afin de croître z , on choisit une variable entrante et une variable sortante :
- la variable entrante (instructions 3 à 7 de l'algorithme 4) : on choisit une variable hors-base a un coefficient positif dans la fonction objectif. Dans notre cas, nous choisissons la variable x_3 (au coefficient maximum), tout en gardant x_1 , x_2 et x_4 à 0.
 - la variable sortante : on choisit une variable de base après le calcul des contraintes sur l'augmentation de la variable x_3 . Ces contraintes sont comme suit :

$$x_5 \geq 0 \Rightarrow 42 - 2x_1 + 4x_2 + 5x_3 + 7x_4 \geq 0$$

$$x_6 \geq 0 \Rightarrow 17 - x_1 + x_2 + 2x_3 + 2x_4 \geq 0$$

$$x_7 \geq 0 \Rightarrow 24 - x_1 + 2x_2 + 3x_3 + 3x_4 \geq 0$$

Ce qui impose :

$$x_3 \leq 8.4, \text{ pour la variable } x_5$$

$$x_3 \leq 8.5, \text{ pour la variable } x_6$$

$$x_3 \leq 8, \text{ pour la variable } x_7$$

La variable x_7 est désignée comme variable sortante puisque c'est la première variable à s'annuler d'après les trois contraintes (instruction 9 de l'algorithme 4).

— On obtient le nouveau dictionnaire :

$$\begin{array}{r} x_3 = 8 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - x_4 - \frac{1}{3}x_7 \\ x_5 = 2 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - 2x_4 - \frac{5}{3}x_7 \\ x_6 = 1 - \frac{1}{3}x_1 + \frac{1}{3}x_2 + \frac{5}{3}x_7 \\ \hline z = 144 + x_1 - 3x_2 - x_4 - 6x_7 \end{array} \quad D_{II}$$

3. toujours afin de croître z , l'instruction 3 de l'algorithme 4 est exécutée. On choisit une variable entrante et une variable sortante :

— la variable entrante : on choisit la variables x_1 puisqu'elle correspond à la variable hors-base ayant le coefficient positif (généralement maximum dans la fonction objectif).

— la variable sortante : les contraintes sur l'augmentation de la variable x_1 sont :

$$x_3 \geq 0, \text{ ce qui impose } x_1 \leq 24$$

$$x_5 \geq 0, \text{ ce qui impose } x_1 \leq 6$$

$$x_7 \geq 0, \text{ ce qui impose } x_1 \leq 3$$

La variable x_6 est bien désignée comme variable entrante.

— On obtient le nouveau dictionnaire :

$$\begin{array}{r} x_1 = 3 + x_2 - 3x_6 + 2x_7 \\ x_3 = 7 - x_2 - x_4 + x_6 - x_7 \\ x_5 = 1 - x_2 - 2x_4 + x_6 + x_7 \\ \hline z = 147 - 2x_2 - x_4 - 3x_6 - 4x_7 \end{array} \quad D_{III}$$

4. les coefficients de toutes les variables hors-base dans la fonction objectif sont négatif (instructions 5 et 6 de l'algorithme 4), ce qui implique que le dictionnaire D_{III} correspond à la solution optimale avec :

$$(x_1^*, x_2^*, x_3^*, x_4^*) = (3, 0, 7, 0) \text{ et } z = 147$$

Ce qui veut dire que l'entreprise (interprétation de la solution optimale) :

- doit fabriquer chaque jour : 3 cendriers, aucun bol, 7 cruches et aucun vase ;
- consomme toutes les heures de cuisson et de peinture, alors qu'il reste une heure de moulage disponible ;
- fait le chiffre d'affaire de 147.

7.3.1 Forme du pivot

Le formalisme du pivot est un formalisme plus automatique que celui du dictionnaire. Il permet de cacher tout à fait le sens des opérations, pour appliquer le SIMPLEXE à un problème linéaire.

En prenant le PL1 de l'exemple 47, nous obtenons :

Formalisme du dictionnaire (Dictionnaire D_1) :

$$\begin{array}{r}
 x_5 = 42 - 2x_1 - 4x_2 - 5x_3 - 7x_4 \\
 x_6 = 17 - x_1 - x_2 - 2x_3 - 2x_4 \\
 x_7 = 24 - x_1 - 2x_2 - 3x_3 - 3x_4 \\
 z = 7x_1 + 9x_2 + 18x_3 + 17x_4
 \end{array}$$

On choisit x_3 comme variable entrante. Pour déterminer la variable sortante, on cherche quelle est l'inégalité, parmi $x_5 \geq 0, x_6 \geq 0, x_7 \geq 0$, qui contraint le plus x_3 .

Formalisme du pivot (Tableau T_I) :

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_5	2	4	5	7	1	0	0	42 → 8.4
x_6	1	1	2	2	0	1	0	17 → 8.5
x_7	1	2	3	3	0	0	1	24 → 8
Z	7	9	18	17	0	0	0	

On choisit, dans la dernière ligne, une colonne dont le coefficient est strictement positif et de valeur maximale, ce qui correspond à la troisième colonne.

La colonne de x_3 s'appelle *colonne pivot*. Pour tout élément strictement positif r de la colonne pivot (sauf celui de la dernière ligne) on calcule le rapport s/r (les valeurs en rouge) où s est l'élément de la dernière colonne sur la même ligne que r . La ligne pour laquelle ce rapport est le plus petit s'appelle *ligne pivot* (ici la 3^{ème}). On appelle l'élément issu de l'intersection de la ligne pivot et de la colonne pivot le *pivot*.

Pour calculer le nouveau tableau SIMPLEXE T_{II} , on applique la règle du pivot (ou du rectangle) donnée par la définition 81 sur le tableau initial T_I .

Définition 81 (règle du pivot) Soient les valeurs a, b, c et d qui représentent les quatre coins du rectangle, avec a la valeur *pivot* (égale à a). L'application de la règle du pivot sur les quatre valeurs se donne comme suit :

$$\begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline 1 & \frac{b}{a} \\ \hline 0 & d - \frac{b \cdot c}{a} \\ \hline \end{array} \quad (21)$$

où on remplace :

- le pivot a par la valeur 1,
- chaque valeur b sur la même ligne du pivot par $\frac{b}{a}$,
- chaque valeur c sur la même colonne du pivot par la valeur 0,

- chaque valeur d sur la même diagonale du pivot par $d - \frac{b \cdot c}{a}$, avec :
 - b : intersection de la ligne de a (la ligne pivot) et la colonne de d ,
 - c : intersection de la colonne de a (la colonne pivot) et la ligne de d .

Ainsi, le nouveau tableau SIMPLEXE T_{II} se donne comme suit :

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_5	$\frac{1}{3}$	$\frac{2}{3}$	0	2	1	0	$-\frac{5}{3}$	2
x_6	$\frac{1}{3}$	$-\frac{1}{3}$	0	0	0	1	$-\frac{2}{3}$	1
x_3	$\frac{1}{3}$	$\frac{2}{3}$	1	1	0	0	$\frac{1}{3}$	8
Z	1	-3	0	-1	0	0	-6	-144

T_{II}

On remarque qu'un coefficient positif de valeur maximale figure dans la dernière ligne de T_{II} , par conséquent :

1. on calcule les variables : entrante et sortante, qui correspondent respectivement la *colonne pivot* et la *ligne pivot*,
2. puis en applique à nouveau la règle du rectangle sur T_{II} .

On obtient par la suite le nouveau tableau SIMPLEXE T_{III} :

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_5	0	1	0	2	1	-1	-1	1
x_1	1	-1	0	0	0	3	-2	3
x_3	0	1	1	1	0	-1	1	7
z	0	-2	0	-1	0	-3	-4	-147

T_{III}

Tous les éléments de la dernière ligne sont négatifs, ce qui implique que T_{III} représente la solution optimale.

Dans ce formalisme, la valeur prise par z (notée z^*) quand on annule les variables hors-base est l'opposé de la valeur contenue dans la case qui constitue l'intersection de la dernière ligne et la dernière colonne, ce qui correspond à $z^* = 147$.

7.3.2 Forme matricielle d'un PL

La forme matricielle d'un PL donné par 18, s'exprime comme suit :

- la fonction objectif :

$$\max z = \sum_{j=1}^n c_j x_j$$

s'écrit comme suit :

$$\max z = CX$$

- et les contraintes :

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, \dots, m$$

comme suit :

$$AX = B$$

avec :

$$C = [c_1, c_2, \dots, c_n]$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$B = [b_1, b_2, \dots, b_m]^T$$

$$X = [x_1, x_2, \dots, x_n]^T$$

Ce qui donne la matrice suivante :

$$\begin{bmatrix} A & B \\ C & \end{bmatrix} \quad (22)$$

Par conséquent le PL s'exprime comme suit :

$$\begin{aligned} \max z &= CX \\ \text{s.c. } \begin{cases} AX \leq B \\ X \geq 0 \end{cases} \end{aligned} \quad (23)$$

Exemple 49 (forme matricielle d'un PL) Soit le PL de l'exemple 48.

La forme matricielle du PL 19 se donne comme suit :

$$\max z = [7 \quad 9 \quad 18 \quad 17] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\text{s.c. } \begin{cases} \begin{bmatrix} 7 & 4 & 5 & 7 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \leq \begin{bmatrix} 42 \\ 17 \\ 24 \end{bmatrix} \\ [x_1 \quad x_2 \quad x_3 \quad x_4]^T \geq 0 \end{cases}$$

7.4 LA DÉGÉNÉRESCENCE ET LE CYCLAGE

Pour un PL écrit sous sa forme standard (voir le modèle 18) [Charon et al., 1996] :

- tout n-uplet $(x_1^*, x_2^*, \dots, x_n^*)$ satisfaisant les contraintes constitue une *solution réalisable*,
- une solution réalisable qui maximise la fonction objectif est dite *solution optimale*,
- si le problème n'admet aucune solution réalisable, il est dit *infaisable*,
- si le problème admet des solutions réalisables mais n'a pas de valeur optimale, il est dit *non bornée*,
- une solution que l'on peut exprimer à l'aide d'un dictionnaire est dite *solution de base réalisable*,

- un dictionnaire est un système d'équations linéaires liant $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}$ et z , et satisfaisant les deux propriétés suivantes :
 - les équations constituant un dictionnaire quelconque doivent exprimer z et m des $n + m$ variables x_1, \dots, x_{n+m} (les m variables en base) en fonction des n autres variables (les n variables hors-base);
 - tout dictionnaire est algébriquement équivalent au dictionnaire définissant les variables d'écart et la fonction objectif (dictionnaire initiale), c'est-à-dire au dictionnaire :

$$\begin{cases} x_{n+i} = b_i - \sum_{j=1}^n a_{ij}x_j & \text{pour } i = 1, \dots, m \\ z = \sum_{j=1}^n c_jx_j \end{cases}$$

- dans l'algorithme du SIMPLEXE, on ne considère que des solutions de bases réalisables (ce qui revient à ne considérer que les sommets du polyèdre des contraintes).
- les solutions de base réalisables avec une ou plusieurs variables de base à la valeur 0 sont dites *dégénérées*.
- on dit qu'il y a *cyclage* lorsque au bout d'un nombre fini d'itérations, on retrouve un dictionnaire déjà rencontré, c'est-à-dire avec une même partition des $m + n$ variables en variables de base et variables hors-base. Un problème qu'on peut éviter en appliquant la règle du plus petit indice (*règle de Bland*) donnée par le théorème de Bland 4. Dans ce cas on choisit la variable entrante et la variable sortantes celles ayant le plus petit indice.

Théorème 4 (théorème de Bland) *Il ne peut y avoir cyclage lorsque, à toute itération effectuée à partir d'un dictionnaire dégénéré, on choisit les variables entrante et sortante comme celles du plus petit indice parmi les candidats possibles.*

 DUALITÉ

8.1 INTRODUCTION

Dans ce chapitre nous introduisons brièvement la notion de dualité appliquée sur la forme standard du PL primal.

En premier lieu nous décrivons la procédure de passage d'un PL primal vers le dual. Par la suite nous exposons la résolution du dual et la relation entre le primal et le dual.

8.2 PROBLÈME DUAL

Un programme linéaire du problème primal est caractérisé par son tableau SIMPLEXE donné par la forme matricielle 23 dans la section 7.3.2.

Le problème dual s'obtient en transposant la matrice donnée par 22 :

$$\begin{bmatrix} A^T & C^T \\ B^T & \end{bmatrix} \quad (24)$$

En supposant que les variables de décisions du problème dual noté (D) sont définies par le vecteur colonne :

$$Y = [y_1, y_2, \dots, y_m]$$

Le programme linéaire du problème dual s'écrit :

$$\begin{aligned} \min \omega &= \sum_{i=1}^m b_i y_i \\ \text{s.c} \begin{cases} \sum_{i=1}^m a_{ji} y_i \geq c_j, j = 1, \dots, n \\ y_i \geq 0, i = 1, \dots, m \end{cases} & \quad (D) \end{aligned} \quad (25)$$

Exemple 50 (Représentation graphique d'un problème dual) *Considérons le PL donné via le modèle 17 de l'exemple 47 du chapitre précédent, qu'on reformule avec les variables x_1 et x_2 , ce qui donne :*

$$\begin{aligned} \max z &= 400x_1 + 200x_2 \\ \text{s.c} \begin{cases} 30x_1 + 20x_2 \leq 6000 \\ 40x_1 + 10x_2 \leq 4000 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} & \quad (P_1) \end{aligned} \quad (26)$$

Le dual du PL (P₁) 26 est un PL dont la fonction objectif est une fonction de minimisation, tel qu'il est donné par le PL (D₁) 27 ci-dessous :

$$\begin{aligned} \min \omega &= 6000y_1 + 4000y_2 \\ \text{s.c} \begin{cases} 30y_1 + 40y_2 \geq 400 \\ 20y_1 + 10y_2 \geq 200 \\ y_1 \geq 0, y_2 \geq 0 \end{cases} & \quad (D_1) \end{aligned} \quad (27)$$

8.3 RÉSOLUTION DU PROBLÈME DUAL

Considérons la proposition ci-dessous :

Proposition 1 [Charon et al., 1996] Soient $(x_1^*, x_2^*, \dots, x_n^*)$ une solution réalisable du problème primal et $(y_1^*, y_2^*, \dots, y_m^*)$ une solution réalisable du problème dual. On a :

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i$$

De plus, si les deux quantités ci-dessus sont égales, alors $x_1^*, x_2^*, \dots, x_n^*$ constituent une solution optimale du problème primal et $y_1^*, y_2^*, \dots, y_m^*$ une solution optimale du problème dual.

Nous constatons que dans le cas où le problème primal est sous sa forme standard (voir le modèle 18), le problème dual ne le sera pas. L'application du SIMPLEXE sur le problème dual impose le passage du dual, exprimé par le système linéaire 25, vers la forme standard. Ce qui donne le PL équivalent, noté (D').

$$\begin{aligned} \max -\omega &= \sum_{i=1}^m -b_i y_i \\ \text{s.c} \begin{cases} \sum_{i=1}^m -a_{ji} y_i \leq -c_j, j = 1, \dots, n \\ y_i \geq 0, i = 1, \dots, m \end{cases} & \quad (D') \end{aligned} \quad (28)$$

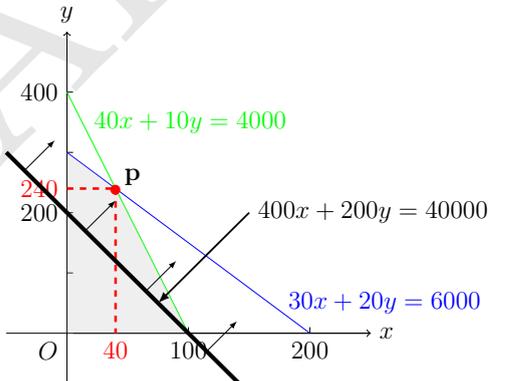
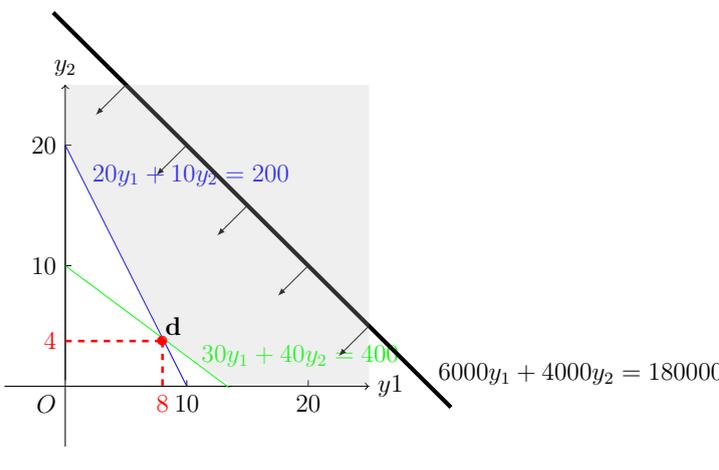
Théorème 5 (Théorème de la dualité) [Charon et al., 1996] Si le primal a une solution optimale $x_1^*, x_2^*, \dots, x_n^*$, alors le problème dual a une solution optimale $y_1^*, y_2^*, \dots, y_m^*$ et $\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*$.

Le théorème 5 nous permet d'affirmer que l'optimum (s'il existe) du problème primal est égal à celui du problème dual. Cet optimum représente le maximum de la fonction objectif du PL (problème primal) et le minimum du dual (D) (problème dual-modèle 25).

Le relation de l'optimum : $z = \sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i y_i$ induit que y_i doit représenter la valeur unitaire de la ressource i . Ces variables duales y_i sont souvent appelées *prix implicite*. Leur valeur donne le montant maximum que l'on serait prêt à payer pour obtenir une unité supplémentaire de la ressource i .

Exemple 51 (Représentation graphique d'un problème dual) *Considérons le PL (P_1) de l'exemple 47 du chapitre précédent. Dans le tableau 6 nous présentons le problème primal en fonction de x_1 et x_2 et son dual en fonction de y_1 et y_2 et leurs résolutions graphiques respectives.*

TABLE 6 – Table de comparaison entre le problème primal et le problème dual

PL primal (P_1)	Résolution graphique de (P_1)
$\begin{aligned} \max z &= 400x_1 + 200x_2 \\ \text{s.c} \quad &\begin{cases} 30x_1 + 20x_2 \leq 6000 \\ 40x_1 + 10x_2 \leq 4000 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned} \quad (P_1)$	
PL dual (D_1)	Résolution graphique de (D_1)
$\begin{aligned} \min \omega &= 6000y_1 + 4000y_2 \\ \text{s.c} \quad &\begin{cases} 30y_1 + 40y_2 \geq 400 \\ 20y_1 + 10y_2 \geq 200 \\ y_1 \geq 0, y_2 \geq 0 \end{cases} \end{aligned} \quad (D_1)$	

Comme nous pouvons le constater :

- dans le cas de (P_1) la droite représentant la fonction objectif est déplacée vers le haut pour atteindre l'optimum (le maximum),
- alors que dans le cas de (D_1) la droite représentant le dual de (P_1) est déplacée vers le bas pour atteindre l'optimum (le minimum).

D'après les deux résolutions graphiques nous avons, conformément au théorème 5, $\omega^* = z^* = 64000$ avec :

$(x_1^*, x_2^*) = (40, 240)$ solution optimale de (P_1), exprimée par le point p.

et $(y_1^*, y_2^*) = (8, 4)$ solution optimale de (D_1), exprimée par le point d.

Concernant la solution optimale du dual $(y_1^*, y_2^*, \dots, y_n^*)$ peut être déduite de la solution optimale du primal notée $(x_1^*, x_2^*, \dots, x_n^*)$ et vice versa (voir la proposition 2).

Proposition 2 Si le problème primal admet une solution optimale et si l'expression de la fonction objectif du primal dans le dernier dictionnaire obtenu par la méthode du simplexe s'écrit :

$$z = z^* + \sum_{k=1}^{n+m} d_k x_k, \quad x_{n+i} : \text{la } i^{\text{ème}} \text{ variable d'écart,}$$

alors une solution optimale du problème dual est donnée par $y_i^* = -d_{n+i}$.

Exemple 52 (solution optimale primal-dual) Soit le problème traité dans l'exemple 48 cité dans le chapitre précédent.

Le dernier tableau obtenu via la forme du pivot était :

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_5	0	1	0	2	1	-1	-1	1
x_1	1	1	0	0	0	3	-2	3
x_3	0	1	1	1	0	-1	1	7
z	0	-2	0	-1	0	-3	-4	-147

T_{III}

Sachant que le dual du dual est le primal, on applique la proposition 2 sur notre problème, ce qui donne :

- les valeurs en gras sur la verticale donne la solution optimale du primal : $(x_1^*, x_2^*, x_3^*, x_4^*) = (3, 0, 7, 0)$,
- les valeurs en gras sur l'horizontale donne la solution optimale du dual : $(y_1^*, y_2^*, y_3^*) = (0, 3, 4)$,
- et l'optimum est $z^* = \omega^* = 147$

BIBLIOGRAPHIE

- [Berge, 1985] BERGE, C. (1985). Graphs and Hypergraphs. Elsevier Science Ltd., Oxford, UK, UK.
- [Bondy et Murty, 2007] BONDY, J.-A. et MURTY, U. S. R. (2007). Graph theory. Graduate texts in mathematics. Springer, New York, London. OHX.
- [Charon et al., 1996] CHARON, I., GERMA, A. et HUDRY, O. (1996). Méthodes d'optimisation combinatoire. Collection pédagogique de télécommunication. Masson.
- [Gondran et Minoux, 1995] GONDRAN, M. et MINOUX, M. (1995). Graphes et algorithmes. Collection De la Direction des études et recherches d'Électricité de France. Eyrolles.
- [Hennecart et al., 2012] HENNECART, F., BRETTO, A. et FAISANT, A. (2012). Éléments de théorie des graphes. IRIS. SPRINGER.
- [Labelle, 1981] LABELLE, J. (1981). Théorie des graphes. Modulo.
- [Müller, 2011] MÜLLER, D. (2011). Cours : Introduction à la théorie des graphes. <https://www.apprendre-en-ligne.net/graphes/graphes.pdf>. CAHIERS DE LA COMMISSION ROMANDE DE MATHÉMATIQUE (CRM).
- [Prins, 1994] PRINS, C. (1994). Algorithmes de graphes : avec programmes en Pascal. Eyrolles.
- [Thulasiraman, 2003] THULASIRAMAN, K. (2003). The Circuits and Filters Handbook. CRC Press, Inc., Boca Raton, FL, USA, 2nd édition.
- [Winston, 1991] WINSTON, W. (1991). Introduction to mathematical programming : applications and algorithms. Business Statistics. PWS-Kent Publishing Company.