Introduction

La recherche opérationnelle (RO) peut se définir comme la mise en œuvre de méthodes scientifiques, essentiellement mathématiques, en vue de prendre la meilleure décision possible. Elle fournit des outils pour rationaliser, simuler et optimiser l'architecture et le fonctionnement des systèmes industriels et économiques. Elle propose des modèles pour analyser des situations complexes et permet aux décideurs de faire des choix efficaces et robustes.

La RO est une discipline scientifique dont l'utilité et les champs d'exploitation n'ont fait que croître au fil des décennies, aidée en cela par les progrès dans les mathématiques et les techniques d'optimisation, ainsi que par l'explosion des possibilités offertes par l'informatique.

L'histoire de la recherche opérationnelle est intimement liée à la programmation linéaire. En effet, c'est à la suite de la création de la première méthode de résolution efficace de programmation linéaire, à savoir l'algorithme du simplexe en 1947, que s'est développée la RO. Cette méthode consiste à minimiser (ou maximiser) une fonction linéaire sous des contraintes également linéaires, ce qui, en pratique, permet de modéliser un grand nombre de situations.

Au côté de la programmation linéaire figure la théorie des graphes qui offre également différentes possibilités de modélisation : plus courts chemins, arbres couvrants de poids minimum, flots de transports, contrôle des programmes....

Chapitre 1 : Programmation linéaire

Un problème résolu en termes de la recherche opérationnelle nécessite généralement les étapes suivantes :

Problème $\Rightarrow$ Modèle $\Rightarrow$ Méthode $\Rightarrow$ Solution

Premièrement il faut modéliser le problème en le transformant en un modèle théorique (programme linéaire, graphe, arbre...), par la suite il faut déterminer la nature du problème (problème linéaire de maximisation, problème de coloration, problème du plus court chemin...) et la méthode à adopter pour le résoudre (méthode du *Simplexe*, algorithme de *Dijkstra*, algorithme de *Ford-Fulkerson...*), et finalement il faut utiliser la méthode choisie pour proposer une solution au problème en question et cela avant de prendre des décisions sur les futures stratégies d'organisation.

1. Aspect géométrique de la programmation linéaire

Nous commençons par un exemple qui peut être résolu en termes de programmes linéaires (PL).

1.1. Exemple d'un problème de maximisation

Exemple 1.1

Une entreprise peut fabriquer, sur une machine donnée, travaillant 45heurs/semaine 3 produits différents P_1, P_2, et P_3 :

- La machine peut fabriquer un seul type de produits à la fois.
- Les temps de réglage de la machine sont négligeables.
- Une unité de produit P_1 laisse un profit net de 4 ($\times$10dhs).
- Une unité de produit P_2 laisse un profit net de 12 ($\times$10dhs).
- Une unité de produit P_3 laisse un profit net de 3 ($\times$10dhs).
- Le rendement de la machine pour le produit P_1 et 50 unité/h.
- Le rendement de la machine pour le produit P_2 et 25 unité/h.
- Le rendement de la machine pour le produit P_3 et 75 unité/h.

L'étude du marché montre que les possibilités de vente ne dépassent pas :

- 1000 unités de P_1 par semaine.
- 500 unités de P_2 par semaine.
- 1500 unités de P_3 par semaine.

Le but de ce problème est de répartir la capacité de production entre les 3 produits de manière à maximiser le profit hebdomadaire.

1.2. Formulation algébrique

On pose le problème précédent sous forme algébrique:

Variables-inconnues

x_1, x_2, x_3 sont respectivement les quantités des produits P_1, P_2, et P_3 qui doivent êtres fabriquées.

Contraintes

- Les quantités des produits P_1, P_2, et P_3 ne doivent pas dépasser, respectivement 1000, 500, et 1500 par semaines :

$$x_1 \leq 1000 \quad (1)$$
$$x_2 \leq 500 \quad (2)$$
$$x_3 \leq 1500 \quad (3)$$

- La somme des temps de fabrication ne doit pas dépasser la disponibilité totale de la machine

$$3x_1 + 6x_2 + 2x_3 \leq 6750 \qquad (4)$$

- Les quantités x_1, x_2, et x_3 sont positives ou nulles :

$$x_1 \geq 0 \qquad (5)$$
$$x_2 \geq 0 \qquad (6)$$
$$x_3 \geq 0 \qquad (7)$$

Les inégalités (1), (2), (3), (4), (5), (6) et (7) constituent les contraintes du problème.

Maximisation du profit

L'objectif du problème est choisir x_1, x_2, et x_3 de manière que le profit hebdomadaire soit maximal.

Le profit : $4x_1 + 12x_2 + 3x_3$ ($\times$10dh)

On écrit : $[\max]\ Z = 4x_1 + 12x_2 + 3x_3$ (8)

Remarque

- Cette fonctionnelle est appelée la fonction économique.
- Cette fonction est linéaire.

1.3. Signification géométrique

Cas de 2 variables (n=2)

Etude d'un sous-problème du problème précédent :

- Elimination du produit P_3 de la gamme.
- Réduction de l'activité de l'atelier à 35 heures/semaine

  Le problème devient :

$$\begin{aligned} &x_1 \leq 1000 \\ &x_2 \leq 500 \\ &x_1 + 2x_2 \leq 1750 \\ &x_1, x_2 \geq 0 \\ &[\max]\ Z = 4x_1 + 12x_2 \end{aligned}$$

Dans ce cas il est possible de recourir à la méthode géométrique en représentant le problème dans l'espace à 2 dimensions (le plan) avec les axes (Ox_1) et (Ox_2) : Toute solution du problème est représentée par un point (x_1, x_2) du plan (fig.1.1).

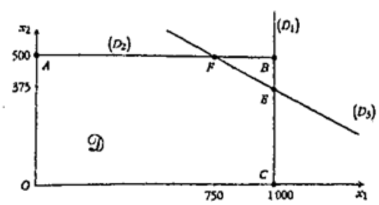


Figure 1.1

On représente les droites (D_1), (D_2), et (D_3) : $\begin{aligned}(D_1) &: x_1 = 1000 \\ (D_2) &: x_2 = 500 \\ (D_3) &: x_1 + 2x_2 = 1750\end{aligned}$

Les contraintes $0 \leq x_1 \leq 1000$, $0 \leq x_2 \leq 500$ et $x_1 + 2x_2 \leq 1750$ sont représentées par le domaine *(D)* délimité par le polygone convexe *OAFEC* (fig.1.1). Les points associés aux solutions admissibles se trouvent à la frontière et à l'intérieur de ce polygone. Il faut choisir parmi les solutions admissibles une solution qui soit optimale, c'est-à-dire qui maximise la fonction économique : $Z = 4x_1 + 12x_2$

L'ensemble des points : $\Delta_0 = \{M(x,y) \ / \ Z = 0\}$ est la droite $\Delta_0 : x_2 = -\dfrac{1}{3}x_1$

(Δ_0) n'a qu'un seul point en commun avec le domaine admissible : c'est le point $O(0,0)$:

(Production =0 ⇒ Bénéfice=0)

L'ensemble des points tels que $Z=4000$ est la droite Δ_{4000} parallèle à Δ_0 et passant par le point $C(x_1=1000,\ x_2=0)$.

On donne à Z une valeur arbitraire α $(Z=\alpha)$: $d(O,(\Delta_\alpha)) = \overline{OH} = \dfrac{\alpha}{\sqrt{4^2+12^2}}$

La valeur α de la fonction économique est proportionnelle à $\overline{OH}$.

Pour maximiser Z, il suffit de tracer une droite parallèle à (Δ_0) dont la distance à l'origine soit la plus grande possible. La droite en question doit avoir au moins un point en commun avec le polygone $OAFEC$ des solutions admissibles (fig.1.2).

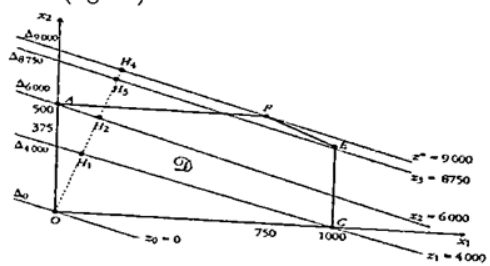


Figure 1.2

- La droite optimale est obtenue pour $\alpha=9000$.
- Δ_{9000} a un seul point en commun avec le polygone (le point F).
- La solution optimale :

$$x_1^* = 750$$
$$x_2^* = 500$$
$$Z^* = 90000 \text{ dhs}$$

Remarque

- La méthode géométrique permet de résoudre tous les programmes linéaires à *2* variables.
- Le principe est de déplacer la droite de la fonction économique parallèlement à une position initiale évidente (le point O par exemple) tant que l'intersection avec le domaine des solutions admissibles n'est pas vide.
- La résolution s'arrête lorsque la droite couvrante n'a plus qu'un seul point de contact (ou une arête de contact) avec le polygone D $(D \neq \varnothing)$.

Cas de 3 variables (n=3)

Dans ce cas il est possible de recourir à la méthode géométrique en représentant le problème dans l'espace à 3 dimensions avec les axes (Ox_1), (Ox_2), et (Ox_3). Toute solution du problème est représentée par un point $(x_1,\ x_2,\ x_3)$ de l'espace.

Les contraintes $x_1, x_2, x_3 \geq 0$ signifient que le domaine recherché est inclus dans l'octant positif de ce système de coordonnées.

On représente les plans P_1, P_2, et P_3 : $\begin{aligned} P_1 &: x_1 = 1000 \\ P_2 &: x_2 = 500 \\ P_3 &: x_3 = 1500 \end{aligned}$

Les contraintes $0 \leq x_1 \leq 1000$, $0 \leq x_2 \leq 500$ et $0 \leq x_3 \leq 1500$ sont représentées par le parallélépipède $OABCDEFG$ (fig.1.3). Les solutions admissibles se trouvent à la surface et l'intérieur de ce parallélépipède.

En introduisant le plan $P_4 : 3x_1 + 6x_2 + 2x_3 = 6750$ On vérifie que les points-solutions doivent nécessairement se trouver à l'intérieur ou à la surface du tétraèdre $OA'B'C'$ (fig.1.3). Le tétraèdre $OA'B'C'$ est formé par les 3 plans de cordonnées et le plan P_4.

Les intersections de P_4 avec les axes (Ox_1), (Ox_2), (Ox_3) sont respectivement A', B' et C'. Le plan P_4 coupe les droites (FG), (GD) et (GB) respectivement aux points P, Q et R (fig.1.3).

Le domaine des solutions admissibles de notre problème est le solide commun au parallélépipède $OABCDEFG$ et le tétraèdre $OA'B'C'$. Les points associés aux solutions admissibles se trouvent à la surface et à l'intérieur du polyèdre $OABCDEFPQR$ (fig.1.3).

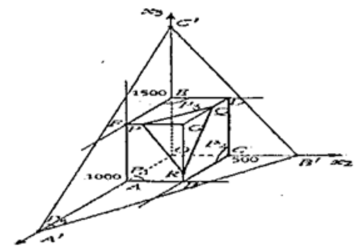


Figure 1.3

Une solution optimale est une solution qui permet de maximiser la fonction économique :

$$Z = 4x_1 + 12x_2 + 3x_3$$

L'équation $Z = 4x_1 + 12x_2 + 3x_3$ constitue une famille de plans parallèles P_Z.

Considérons par exemple le plan $P_{(Z=6000)}$ passant par le point $C(0,500,0)$ et dont l'équation est :

$$4x_1 + 12x_2 + 3x_3 = 6000$$

Le plan $P_{(Z=6000)}$ est le plan ICJ (fig.1.4) tel que : $I(1500,0,0)$,$C(0,500,0)$ et $J(0,0,2000)$.

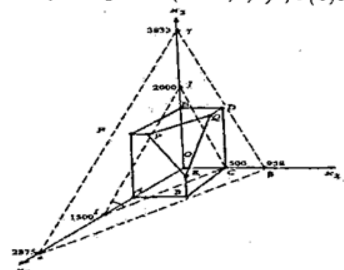


Figure 1.4

On a : $d(O,P_Z) = \dfrac{Z}{\sqrt{4^2 + 12^2 + 3^2}} = \overline{OH}$

$$\overline{OH} = \frac{Z}{13} \Leftrightarrow Z = 13.\overline{OH}$$

Pour maximiser Z il suffit de tracer un plan parallèle au plan ICJ et dont la distance à l'origine soit la plus grande possible. Le plan recherché doit avoir au moins un point commun avec le polyèdre $OABCDEFPQR$.

La solution optimale est obtenue pour le plan $P_{(Z=11500)}$ parallèle au plan ICJ et passant par le point $Q(250,500,1500)$ (fig.1.4) : $Z^* = 11500$ pour $x_1^* = 250, x_2^* = 500, et\ x_3^* = 1500$

Remarque

- Aux sommets B et P : $Z=10000$
- Au sommet R : $Z=11125$
- Au sommet G : $Z=14500$ (point exclu)

1.4. Raisonnement économique

Ce type de raisonnement consiste à transformer les rendements horaires en rendements en termes d'unités monétaires :

$$P_1 : 200\ (\times 10dh)\ /\ h$$
$$P_2 : 300\ (\times 10dh)\ /\ h$$
$$P_3 : 225\ (\times 10dh)\ /\ h$$

Maximiser le profit revient à fabriquer la plus grande quantité du produit P_2 (P_2 a le profit monétaire le plus élevé). S'il reste du temps, on fabriquera des unités de P_3. Finalement si le temps de production n'est pas épuisé, il faudra produire des unités de P_1.

Si on fabrique les quantités maximales de tous les objets on fera fonctionner la machine *60h*. Cela n'est pas possible, ce qui nécessite une meilleure distribution de la fabrication.

Dans cet exemple on doit fabriquer toutes les unités de P_2 (on occupe la machine 20h). Puis on fabrique toutes les unités de P_3 (on occupe la machine 20h). Il reste 5h pour fabriquer les unités de P_1.

On aura : P_1 : 250 unités, P_2 : 500 unités et P_3 :1500 unités.

Ce qui donne un profit total de: $(250\times4) + (500\times12) + (1500\times3) = 11500\ (\times 10dh)$ /semaine

Remarque

- La méthode économique est simple pourtant elle n'a pas un caractère général.
- L'objectif est d'introduire un algorithme permettant la résolution générale des programmes linéaires.

1.5. Difficultés de généralisation

Limites de la méthode géométrique

La méthode géométrique ne peut pas s'étendre au cas d'un nombre de variables > 3 (la représentation géométrique n'est pas possible dans un espace à n dimensions dés que n dépasse *3*). Un autre problème se pose lorsque le nombre de contraintes augmente même pour trois variables la représentation géométrique devient non efficace.

Caractérisation d'une solution optimale d'un PL

Théorème

Si l'ensemble des contraintes d'un programme linéaire forme un polyèdre non vide, il existe une solution optimale qui est un sommet de ce polyèdre.

Remarque

- L'ensemble de contraintes d'un PL ne forme pas toujours un polyèdre convexe : il peut arriver que cet ensemble comporte des points à l'infini et donc forme un polytope convexe ouvert (le polyèdre est un polytope fermé).
- Dans ce cas, bien que des solutions réalisables du PL existent, il n'existe pas nécessairement de solution optimale : Les solutions optimales peuvent être rejetées à l'infini.
- Si une solution optimale existe alors il existe une solution optimale située en un sommet du polytope.
- Si les contraintes sont contradictoires alors le polyèdre est vide et le PL est impossible.

En se basant sur ce théorème Il suffit de déterminer les cordonnées de tous les sommets du polyèdre est de calculer la valeur de la fonction économique qui correspond à chacun d'eux, et de choisir la plus grande de ces valeurs. Avec n variable et m contraintes le nombre de points d'intersections est de l'ordre :

$$C_{n+m}^{n} = \frac{(n+m)!}{n!m!}$$

Pour $n=15$ et $m=10$, On a 3268760 points d'intersection. Avec un PL de taille industrielle l'énumération de tous les sommets est impraticable si n et m dépassent 20 : Le temps d'exécution de la méthode avec un ordinateur très puissant dépasse des milliards d'années.

Vers le simplexe

La méthode du simplexe a un fondement purement géométrique : elle consiste, en disposant d'un point de départ, qui est un sommet du polyèdre, supposé connu, lors de toute itération à passer d'un sommet M à un sommet voisin M' (décrire une arête du polyèdre) en lequel la valeur de la fonction économique est meilleure (ou au moins aussi bonne) qu'en M. Lorsqu'on atteint un sommet Q pour lequel aucun sommet voisin n'est meilleur, alors l'algorithme s'arrête et le sommet Q est optimal.

2. Méthode du simplexe

2.1. Méthode algébrique du simplexe

Cette méthode consiste à ramener le PL à une forme « standard »:

- Toutes les contraintes sont en égalités.
- Les seconds membres sont positifs.
- La fonction Z est à maximiser.

Sur le PL de l'exemple 1.1 : $x_1 \leq 1000$ est équivaut à : $x_1 + x_4 = 1000$ et $x_4 \geq 0$

- La variable x_4 est nommée «variable d'écart».
- x_4 représente l'écart à la saturation du marché en produit P_1.

En introduisant des variables d'écart x_4, x_5, x_6 et x_7 on aura le système d'équations suivantes :

$$\begin{cases} x_1 \quad + x_4 = 1000 \\ x_2 \quad + x_5 = 500 \\ x_3 \quad + x_6 = 1500 \\ 3x_1 + 6x_2 + 2x_3 + x_7 = 6750 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \end{cases}$$

Remarque

- La variable x_7 représente: $150\times$(le nombre d'heures de travail par semaine non employées).
- Les variables d'écart peuvent être notées $x_{\bar{1}}, x_{\bar{2}}, x_{\bar{3}}$ et $x_{\bar{4}}$.
- Si on a une contrainte en sens inverse, par exemple $2x_1 + x_2 + 5x_3 \geq 1000$, on retrancherait une variable d'écart x_8 pour obtenir $2x_1 + x_2 + 5x_3 - x_8 = 1000$ et $x_8 \geq 0$.
- Les variables d'écart ont une contribution nulle à la fonction économique.

Une solution initiale à ce système correspond au sommet $O(x_1=0,\ x_2=0,\ x_3=0)$:

- Fabrication nulle ⇒ Bénéfice nul.
- Solution admissible au sens mathématique.
- Les m variables ajoutées ($m=4$) x_4, x_5, x_6 et x_7 sont nommées variables de base en O.
- Les variables nulles au sommet O : x_1, x_2 et x_3 sont nommées variables hors-base en O.

On exprime les variables de base en O en fonction des variables hors-base en O :

$$O : \begin{cases} x_4 = 1000 - x_1 \\ x_5 = 500 - x_2 \quad (*) \\ x_6 = 1500 - x_3 \\ x_7 = 6750 - 3x_1 - 6x_2 - 2x_3 \\ Z = 0 + 4x_1 + 12x_2 + 3x_3 \end{cases}$$

Augmenter la fonction $Z=4x_1+12x_2+\ 3x_3$ revient à accroitre la valeur de l'une des variables hors-base: On cherche à accroitre la variable x_2 qui a le coefficient (bénéfice marginal) le plus élevé tout en gardant les autres variables hors-base nulles pour cette itération.

On pose $x_2=\theta$ où θ est un paramètre positif croissant.

Le système devient :

$$\begin{cases} x_4 = 1000 \\ x_5 = 500 - \theta \\ x_6 = 1500 \\ x_7 = 6750 - 6\theta \\ Z = 0 + 12\theta \end{cases}$$

Il faut accroitre θ tout en respectant les contraintes de non-négativité des variables.

On pose donc $\theta=500$.

$$\begin{cases} x_4 = 1000 \\ x_5 = 0\ , \quad x_2 = 500 \\ x_6 = 1500 \\ x_7 = 3750 \\ Z = 6000 \end{cases}$$

On retrouve les cordonnées du sommet C : le programme de production qui engendre un bénéfice $Z=6000(\times 10dh)$.

L'objectif est de suivre ce procédé algébrique qui a permis de passer d'un sommet O du polyèdre à un sommet voisin C, en décrivant une arête de ce polyèdre (l'arête OC).

On applique le même procédé : On exprime les variables de base en C en fonction des variables hors base en C :

- x_2 est la variable entrante dans la base (fig.1.5).
- x_5 est la variable sortante de la base (fig.1.5).

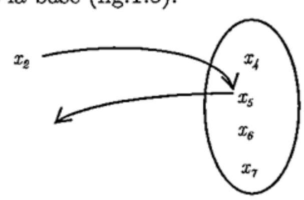


Figure 1.5

Le nouveau système :

$$C : \begin{cases} x_4 = 1000 - x_1 \quad (*) \\ x_2 = 500 - x_5 \\ x_6 = 1500 - x_3 \\ x_7 = 3750 - 3x_1 - 2x_3 + 6x_5 \\ Z = 6000 + 4x_1 + 3x_3 - 12x_5 \end{cases}$$

A partir du sommet C où le bénéfice est *6000* on fait des itérations successives pour accroître ce bénéfice tout en respectant la démarche proposée. On aura à la fin le système suivant:

$$Q : \begin{cases} x_1 = 250 + 2x_5 + \frac{2}{3}x_6 - \frac{1}{3}x_7 \\ x_2 = 500 - x_5 \\ x_4 = 750 - 2x_5 - \frac{2}{3}x_6 + \frac{1}{3}x_7 \\ x_3 = 1500 - x_6 \\ Z = 11500 - 4x_5 - \frac{1}{3}x_6 - \frac{4}{3}x_7 \end{cases}$$

On reconnaît le sommet Q : $x_1=250$, $x_2=500$, $x_3=1500$ avec un bénéfice de $Z=11500$.

Il n'est pas possible d'améliorer Z par ce procédé (tous les coefficients figurant dans Z sont négatifs), le sommet Q est optimal.

L'algorithme du simplexe consiste à progresser d'un sommet initial vers un sommet adjacent en ayant soin de ne pas diminuer la valeur de la fonction économique et de garder toutes les variables non négatives.

Présentons sous forme de tableau le système associé au sommet initial O :

i \ j	A^1	A^2	A^3	A^4	A^5	A^6	A^7	β
4	1	0	0	1	0	0	0	1000
5	0	1	0	0	1	0	0	500
6	0	0	1	0	0	1	0	1500
7	3	6	2	0	0	0	1	6750
	4	12	3	0	0	0	0	$z-0$

- La colonne à gauche est un descriptif de la base : les indices i désignent les lignes de la matrice.
- Chaque indice i est associé à la variable de base x_i.
- La colonne à droite est le vecteur second membre β.

Lors de toute itération :

- L'élément de la matrice situé à l'intersection de la ligne i et de la colonne j sera désigné par α_{ij}.
- L'élément de la colonne β situé à la ligne i sera noté β_i.

Remarque

- Initialement, la colonne β coïncide avec la colonne des seconds membres des contraintes noté b.
- La ligne en bas contient les coefficients de la fonction économique : on l'appelle la ligne des Δ.
- Les variables d'écart sont de profits nuls.

Les formules de changements dus à la sortie d'un vecteur colonne A^s de la base et l'entrée d'un vecteur A^e sont :

- Valeur de la fonction économique :

$$Z' = Z + \frac{\beta_s}{\alpha_{se}} \cdot \Delta_e \text{ où } \alpha_{se} \text{ est nommé pivot.}$$

- Valeurs des variables de base :

$$\beta_k' = \beta_k - \alpha_{ke} \cdot \frac{\beta_s}{\alpha_{se}} \quad (k \neq s) \text{ et } \beta_s' = \frac{\beta_s}{\alpha_{se}}$$

Les β_k' sont les valeurs des variables de base x_k, β_s' est la valeur de la variable entrante x_e.

- Valeurs des éléments d'intersection *ligne k-colonne l*:

$$\alpha_{kl}' = \alpha_{kl} - \alpha_{ke} \cdot \frac{\alpha_{sl}}{\alpha_{se}} \quad (k \neq s) \text{ et } \alpha_{sl}' = \frac{\alpha_{sl}}{\alpha_{se}}$$

Cela signifie qu'on divise la ligne s (celle du pivot) par le pivot α_{se} *($\alpha_{se} > 0$)*

Premier critère de Dantzig

«Pour déterminer la colonne A^e qui doit entrer dans la base, on choisit celle qui comporte le Δ_j positif le plus grand».

Si tous les Δ_j sont négatifs ou nuls : fin, l'optimum est atteint.

Deuxième critère de Dantzig

«Pour déterminer la colonne A^s qui doit sortir de la base, on choisit celle d'indice s tel que $\frac{\beta_k}{\alpha_{ke}}$ soit le plus petit positif $(k=s)$»

Remarque

- Si tous les rapports $\frac{\beta_k}{\alpha_{ke}}$ sont négatifs, alors l'optimum est rejeté à l'infini. Ce cas ne se présente par dans les PL à caractère économique.

- L'optimum sera atteint dès que le premier critère ne sera plus applicable : tous les Δ_j relatifs aux variables hors-base seront négatifs ou nuls.

2.2. Maximisation par la méthode des tableaux

Considérons le tableau associé au système du sommet O :

i	j	A^1	A^2	A^3	A^4	A^5	A^6	A^7	β
4		1	0	0	1	0	0	0	1000
5		0	1	0	0	1	0	0	500
6		0	0	1	0	0	1	0	1500
7		3	6	2	0	0	0	1	6750
		4	12	3	0	0	0	0	$z-0$

On applique les critères de Dantzig:

	c_i	i	j	1	2	3	4	5	6	7	β	$\frac{\beta_i}{\alpha_{ie}}$
	0	4		1	0	0	1	0	0	0	1000	∞
$s=5 \rightarrow$	0	$\boxed{5}$		0	1	0	0	1	0	0	500	500 (*)
	0	6		0	0	1	0	0	1	0	1500	∞
	0	7		3	6	2	0	0	0	1	6750	1125
			c_j	4	12	3	0	0	0	0		
			Δ_j	4	12	3	0	0	0	0	$z-0$	
					$\uparrow e=2$							

On a : $\min_{i\in\{4,5,6,7\}} \dfrac{\beta_i}{\alpha_{i2}} = 500$ pour i=5

On peut appliquer les formules de changements de cordonnées (*e=2, s=5*) :

$$Z' = Z + \frac{\beta_5}{\alpha_{52}}\Delta_2 \qquad \beta_k' = \beta_k - \alpha_{k2}\cdot\frac{\beta_5}{\alpha_{52}} \qquad \alpha_{kl}' = \alpha_{kl} - \alpha_{k2}\cdot\frac{\alpha_{5l}}{\alpha_{52}} \ \ (\mathrm{k} \neq 5) \qquad \alpha_{5l}' = \frac{\alpha_{5l}}{\alpha_{52}} \qquad \beta_5' = \frac{\beta_5}{\alpha_{52}}$$

L'élément α_{52} est le pivot de la transformation.

Dans ce qui suit, on applique les règles pratiques de transformation du tableau :

1) On repère le pivot : l'intersection de la colonne associée à l'indice e (x_e est la variable qui rentre dans la base) et la ligne associée à l'indice s (x_s est la variable qui sort de la base).
2) On divise les éléments de la ligne s par le pivot α_{se}. Par la suite on remplace l'indice du tableau de gauche ,2ème colonne, par l'indice de la colonne qui entre dans la base, aussi que le coefficient c_s du tableau de gauche 1ère colonne par la valeur c_e du coefficient de x_e dans la fonction économique.
3) Parmi les autres lignes du tableau, celles qui comportent un *0* dans la colonne e qui entre dans la base ($\alpha_{ke}=0$ où $k\neq s$), ne sont pas modifiées.
   Les éléments des autres lignes du tableau qui comportent un élément différent de *0* dans la colonne qui entre dans la base ($\alpha_{ke}\neq 0$, où $k\neq s$), sont modifiées comme suit: On multiple les éléments de la nouvelle ligne du pivot (désormais ligne e) par cet élément α_{ke} et on soustrait les résultats aux éléments correspondants de la ligne à modifier.

Par exemple, l'itération 1 permet d'avoir le tableau associé au sommet *C (0,500,0)*:

c_i	i	j	1	2	3	4	5	6	7	β
0	4		1	0	0	1	0	0	0	1000
12	2		0	1	0	0	1	0	0	500
0	6		0	0	1	0	0	1	0	1500
0	7		3	0	2	0	−6	0	1	3750
z		Δ_j	4	0	3	0	−12	0	0	6000

En appliquant les mêmes règles de passage sur ce tableau on arrive à l'itération 4 qui permet d'avoir le tableau associé au sommet Q :

i \ j	1	2	3	4	5	6	7	β	
1	1	0	0	0	-2	$-\frac{2}{3}$	$\frac{1}{3}$	250	
2	0	1	0	0	1	0	0	500	
4	0	0	0	1	2	$\frac{2}{3}$	$-\frac{1}{3}$	750	sommet Q
3	0	0	1	0	0	1	0	1500	
	0	0	0	0	-4	$-\frac{1}{3}$	$-\frac{4}{3}$	$z=-11500$	

L'optimum est atteint en 4 itérations :

- Les Δ_j sont ≤ 0.
- Le bénéfice est de $11500(\times 10\ dh)$ par semaine.

3. Aspect matriciel des programmes linéaires

3.1. Forme standard et écriture matricielle

La forme standard d'un programme linéaire pour la résolution par l'algorithme du simplexe est la suivante :

- Toutes les contraintes explicites sont des équations dont les seconds membres sont des nombres positifs ou nuls.
- Toutes les variables sont ≥ 0.
- La fonction économique Z est à maximiser.

$$F.S: \begin{cases} \sum_{j=1}^{n} a_{ij}x_j = b_i \text{ (où } b_i \geq 0) \text{ pour i=1,2,...,m} \\ x_j \geq 0 \text{ pour j=1,2,...,n} \\ Max\ Z = \sum_{j=1}^{n} c_j x_j \end{cases}$$

Notons :

- $A = (a_{ij})_{i,j}$ la matrice des coefficients des premiers membres des contraintes explicites.
- b le vecteur-colonne des seconds membres de ces contraintes.
- C le vecteur-ligne des coefficients de la fonction économique.
- x le vecteur-colonne des variables du PL.

La forme standard s'écrit matriciellement:

$$F.S: \begin{cases} A \cdot x = b \text{ (où b} \geq 0) \\ x \geq 0 \\ Max\ Z = C \cdot x \end{cases}$$

On note :

A^j la $j^{ème}$ colonne de la matrice A $(j=1,2,\ldots,n)$.

Le système des contraintes explicites peut s'écrire sous forme d'une combinaison linéaire des A^j :

$$x_1 \cdot A^1 + x_2 \cdot A^2 + \cdots + x_n \cdot A^n = b$$

<u>*Règles pratiques pour aller vers la F.S*</u>

- Si l'objectif du PL est de minimiser Z, on le remplace par maximiser Z', où $Z'=-Z$.
- Si le signe d'une variable n'est pas connu à l'avance : par exemple pour $x_j \geq -273$, on pose $x_j' = x_j + 273$ et $x_j' \geq 0$. S'il n'y a pas de borne inférieure pour x_j, on peut poser : $x_j = x_j' - x_j''$ avec $x_j', x_j'' \geq 0$.
- Si une contrainte i comporte un second membre négatif ($b_i<0$), on multiplie par -1 la contrainte.
- Si certaines contraintes sont initialement en inégalité, on peut les ramener à des égalités par introduction de nouvelles variables d'écart:

$$3x_1 + 4x_2 \leq 5 \Leftrightarrow 3x_1 + 4x_2 + x_e = 5 \text{ et } x_e \geq 0$$

$$2x_1 + 7x_2 \geq 11 \Leftrightarrow 2x_1 + 7x_2 - x_e' = 11 \text{ et } x_e' \geq 0$$

Hypothèse sur la F.S

Nous supposons que les coefficients des premiers membres des contraintes explicites constituent une matrice A: $m \times n$ telle que:

- Le nombre de contraintes explicites est inférieur au nombre de variables du système ($m<n$). Cela signifie que si les contraintes explicites sont initialement des inégalités, on introduit m variables d'écart (Nécessairement $m<n'$ ($n'=n+m$)).
- Il est possible d'extraire de la matrice A, m colonnes indépendantes regroupées dans une matrice carrée B ($Dét(B) \neq 0$). Si les contraintes explicites du PL sont initialement des inégalités, les m colonnes de A associées aux variables d'écart forment la matrice:

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

Remarque

- Les 2 hypothèses sont peu restrictives en pratique.
- Le système linéaire $A.x=b$ (m équations et n inconnues) admet dans ces conditions au moins une solution (généralement une infinité).
- Si x' et x'' sont deux solutions différentes de ce système alors toute combinaison linéaire convexe de x' et x'' est une solution de ce système.

Base

Une base d'un PL est un ensemble de m vecteurs-colonne indépendants, extraits de A.

A est une matrice de n colonnes:

$$A = \boxed{A^1 \mid A^2 \mid A^3 \mid \cdots \mid A^n}$$

On extrait de A m colonnes indépendantes d'indices $j_1, j_2, \ldots, j_m$. On associe à la base la matrice carrée $m \times m$ suivante :

$$B = \boxed{A^{j1} \mid A^{j2} \mid \cdots \mid A^{jm}}$$

On note x_B le vecteur-colonne de m variables de base $x_{j1}, x_{j2}, \ldots, x_{jm}$: $x_B \begin{pmatrix} x_{j_1} \\ x_{j_2} \\ \vdots \\ x_{j_m} \end{pmatrix}$

Les (n-m) autres variables sont dites variables hors-base. On note x_N le vecteur–colonne des variables hors base.

Réécriture du PL associé à une base B

On permute les colonnes de la matrice A comme suit :

$$A = \underbrace{\boxed{A^{j1} \mid A^{j2} \mid \cdots \mid A^{jm}}}_{B}\underbrace{\boxed{\quad \mid \quad \mid \cdots \mid \quad}}_{N}$$

Les colonnes à gauche forment la sous matrice de base B et les colonnes à droite forment la sous matrice N des colonnes hors base:

$$A = (B, N) \text{ et } x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$$

Le système de contraintes explicites s'écrit : $B.x_B + N.x_n = b$

Exemple 1.2

$$\begin{cases} 2x_1 + 3x_2 - x_3 + x_4 + 3x_5 = 12 \\ 7x_1 - x_2 + 2x_3 + 5x_4 + x_5 = 6 \end{cases}$$

Sous forme matricielle:

$$A = \begin{array}{ccccc} A^1 & A^2 & A^3 & A^4 & A^5 \\ \hline 2 & 3 & -1 & 1 & 3 \\ 7 & -1 & 2 & 5 & 1 \end{array}$$

$B = (A^2, A^5) = \begin{pmatrix} 3 & 3 \\ -1 & 1 \end{pmatrix}$ forme une base.

$$\begin{pmatrix} 3 & 3 \\ -1 & 1 \end{pmatrix} \times \begin{pmatrix} x_2 \\ x_5 \end{pmatrix} + \begin{pmatrix} 2 & -1 & 1 \\ 7 & 2 & 5 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 12 \\ 6 \end{pmatrix}$$

La solution de base correspond à l'équation $B.x_B=b$. Ce système à m équations et m inconnues et de déterminant non nul a une solution unique : $x_B=B^{-1}.b$ Si $x_B \geq 0$ alors x_B est une solution de base admissible.

Dans l'exemple on a: $\begin{pmatrix} x_2 \\ x_5 \end{pmatrix} = \frac{1}{6} \begin{pmatrix} 1 & -3 \\ 1 & 3 \end{pmatrix} \times \begin{pmatrix} 12 \\ 6 \end{pmatrix}$

Par conséquent $x_B = \begin{pmatrix} -1 \\ 5 \end{pmatrix}$ est une solution de base qui n'est pas admissible.

Théorème

A toute solution de base admissible correspond un sommet du polyèdre des solutions admissibles et un seul.

Remarque

Ce théorème précise la correspondance entre les sommets du polyèdre D et les solutions de base admissibles: il fait le lien entre l'aspect géométrique et l'aspect algébrique en programmation linéaire.
La réciproque de ce théorème n'est vraie qu'en absence de dégénérescence de 2ème espèce : $x_B > 0$ (aucune variable de base n'est nulle).

3.2. Forme matricielle de la fonction économique

Sous forme matricielle : $Z=C.x$ où C est le vecteur-ligne des coefficients initiaux de la fonction économique, et x est le vecteur-colonne des variables du PL.

On pose $C=(C_B, C_N)$ où :

- C_B est le sous vecteur ligne des coefficients associés aux variables de base.
- C_N est le sous vecteur ligne des coefficients associés aux variables hors base.

Sous cette forme on a :

$$Z = C_B \cdot x_B + C_N \cdot x_N$$

Expression de Z en fonction des variables hors base

On a : $x_B = B^{-1} \cdot b - B^{-1} \cdot N \cdot x_N$ (La forme matricielle des variables de base en fonction des variables hors base).

Il s'ensuit que :
$$Z = C_B \cdot B^{-1} \cdot b + (C_N - C_B \cdot B^{-1} \cdot N) \cdot x_N$$
$$\Rightarrow Z = \tilde{Z}_B + \Delta_N \cdot x_N$$

Remarque

- $\tilde{Z}_B$ est une constante qui vaut la valeur numérique de la fonction économique pour la solution de base : $\tilde{Z}_B = C_B \cdot B^{-1} \cdot b$
- $\Delta_N = C_N - C_B \cdot B^{-1} \cdot N$ est le vecteur ligne $1 \times (n\text{-}m)$ des coefficients des variables hors base dans la fonction Z. On a : $\forall j$, $\Delta_j = c_j - C_B \cdot (B^{-1} \cdot N)^j$ où $y^j = (B^{-1} \cdot N)^j$ est la colonne associée à la variable hors base x_j dans la matrice $B^{-1}.N$: $\Delta_j = c_j - \sum c_i.\alpha_{ij}$

4. Algorithme du simplexe: problème de la base initiale

4.1. Cas «favorable»

Toute les m contraintes d'un PL sont initialement sous forme d'inéquations avec p variables principales :

$$\sum_{j=1}^{p} a_{ij} x_j \leq b_i \text{ (où } b_i \geq 0 \text{ et } i = 1,2,..,m)$$

Pour chaque contrainte, on ajoute une variable d'écart : $\sum_{j=1}^{p} a_{ij} x_j + x_{\bar{i}} = b_i$

Les m variables d'écart $x_{\bar{i}}$ forment une base de matrice identité I. Les p variables principales $x_1, x_2, \ldots, x_p$ sont hors-base et correspondent au sommet d'origine O.

L'expression des variables de base en fonction des variables hors-base : $x_{\bar{i}} = b_i - \sum_{j=1}^{p} a_{ij} x_j$,$(i = 1,2,\ldots,m)$

L'expression initiale de la fonction économique est en fonction des variables hors-base :

$$Z = \sum_{j=1}^{p} c_j x_j + (0.x_{\bar{1}} + 0.x_{\bar{2}} + \ldots + 0.x_{\bar{m}})$$

4.2. Cas où une solution est connue à l'avance

Ce procédé est utilisé lorsque le système à optimiser possède déjà un point de fonctionnement: Pour un PL, le point de fonctionnement est une solution admissible vérifiant les $m+n$ contraintes du PL. On vérifie par la suite si cette solution admissible est utilisable pour la résolution par l'algorithme du simplexe : une solution de base est réalisable dans le cas où le nombre de variables strictement positives est au maximum m, et les m colonnes de la matrice A associées à ces variables forment une matrice régulière B.

On cherche l'expression des m variables de base et de Z, en fonction des variables hors-base.

Le PL s'écrit avec la base de matrice B :

$$\begin{cases} B \cdot x_B + N \cdot x_N = b \\ Z = C_B \cdot x_B + C_N \cdot x_N \end{cases}$$

On aura :

$$\begin{cases} x_B = B^{-1} \cdot b - B^{-1} \cdot N \cdot x_N \\ Z = C_B \cdot B^{-1} \cdot b + (C_N - C_B \cdot B^{-1} \cdot N) x_N = \tilde{Z}_B + \Delta_N \cdot x_N \end{cases}$$

Le tableau associé s'écrit (sous forme matricielle) :

$$\begin{cases} I \cdot x_B + B^{-1} \cdot N \cdot x_N = B^{-1} \cdot b \\ 0 \cdot x_B + \Delta_N \cdot x_N = Z - \tilde{Z}_B \end{cases}$$

4.3. Cas d'une base «évidente»

Exemple 1.3

Soit le PL suivant:

$$\begin{cases} y_1 + 3y_4 - y_{\bar{1}} = 4 \\ y_2 + 6y_4 - y_{\bar{2}} = 12 \\ y_3 + 2y_4 - y_{\bar{3}} = 3 \\ y_1, y_2, y_3, y_4, y_{\bar{1}}, y_{\bar{2}}, y_{\bar{3}}, y_{\bar{4}} \geq 0 \\ -[1000y_1 + 500y_2 + 1500y_3 + 750y_4] = Z \text{ [max]} \end{cases}$$

Les variables d'écart $y_{\bar{1}}, y_{\bar{2}}, y_{\bar{3}}$ forment une base de matrice *B'=-I* : Cette base n'est pas admissible. Les variables y_1, y_2 et y_3 forment une base « évidente » de matrice *B=I.*

Soit le PL suivant :

$$\begin{cases} 3y_1 + y_2 \geq 30 \\ y_2 + 2y_3 \geq 24 \\ y_1, y_2, y_3 \geq 0 \\ 16y_1 + 27y_2 + 10y_3 = Z \ [Min] \end{cases}$$

Le PL devient :

$$\begin{cases} 3y_1 + y_2 - y_{\bar{1}} = 30 \\ y_2 + 2y_3 - y_{\bar{2}} = 24 \\ y_1, y_2, y_3, y_{\bar{1}}, y_{\bar{2}} \geq 0 \\ -16y_1 - 27y_2 - 10y_3 = Z' \ [Max] \end{cases}$$

On cherche une base évidente :

$$\begin{cases} y_1 + \frac{1}{3}y_2 - \frac{1}{3}y_{\bar{1}} = 10 \\ \frac{1}{2}y_2 + y_3 - \frac{1}{2}y_{\bar{2}} = 12 \end{cases}$$

- Les variables y_1, y_3 forment une base évidente *B=I.*
- *Z'* en fonction des variables hors base :

$$Z' = -312 - \frac{50}{3}y_2 - 16y_{\bar{1}} - 10y_{\bar{2}}$$: Cette base est l'optimum.

4.4. Cas général: emploi de variables artificielles

Un PL peut comporter 3 types de contraintes *(P)*,*(Q)* et *(R)* :

- Contrainte de type *(P)*: $\sum_{j=1}^{p} a_{ij}x_j \leq b_i$ (où $b_i \geq 0$)

  En introduisant une variable d'écart : $\sum_{j=1}^{p} a_{ij}x_j + x_{\bar{i}} = b_i$

- Contraintes de type *(Q)* : $\sum_{j=1}^{p} a_{kj}x_j \geq b_k$ (où $b_k \geq 0$)

  En introduisant une variable d'écart : $\sum_{j=1}^{p} a_{kj}x_j - x_{\bar{k}} = b_k$

- Contraintes de type *(R)* : $\sum_{j=1}^{p} a_{ij}x_j = b_i$ (où $b_i \geq 0$)

  Ces contraintes ne contiennent pas de variables d'écart.

Exemple 1.4

Soit le système suivant :

$$\begin{cases} x_1 + 2x_2 + x_3 \leq 5 \\ 2x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 = 4 \\ x_1, x_2, x_3 \geq 0 \\ 4x_1 + 5x_2 + 3x_3 = Z \text{ [max]} \end{cases}$$

En introduisant 2 variables d'écart, on aura :

$$\begin{cases} x_1 + 2x_2 + x_3 + x_{\bar{1}} = 5 \\ 2x_1 + x_2 + x_3 - x_{\bar{2}} = 1 \\ x_1 + x_2 = 4 \\ x_1, x_2, x_3, x_{\bar{1}}, x_{\bar{2}} \geq 0 \\ 4x_1 + 5x_2 + 3x_3 + 0x_{\bar{1}} + 0x_{\bar{2}} = Z \text{ [max]} \end{cases}$$

- La colonne associée à $x_{\bar{1}}$ est unitaire : $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$
- On note x_a et x_b les 2 variables artificielles nécessaires pour constituer une base de matrice $B=I$.

$$\begin{cases} x_1 + 2x_2 + x_3 + x_{\bar{1}} = 5 \\ 2x_1 + x_2 + x_3 - x_{\bar{2}} + x_a = 1 \\ x_1 + x_2 + x_b = 4 \\ x_1, x_2, x_3, x_{\bar{1}}, x_{\bar{2}}, x_a, x_b \geq 0 \end{cases}$$

- Les variables artificielles sont ≥ 0.
- Si $x_a \neq 0$ (respectivement $x_b \neq 0$) alors $2x_1 + x_2 + x_3 - x_{\bar{2}} \neq 1$ (respectivement $x_1 + x_2 \neq 4$). Cela signifie qu'une variable artificielle strictement positive entraîne que la contrainte correspondante est violée.

Remarque

- Si toutes les contraintes étaient de type *(P)*, on serait dans le cas « favorable ».
- Pour constituer une base initiale de matrice *B=I*, on peut inclure dans cette base les variables d'écart ajoutées dans les contraintes de type *(P)*, puis compléter cette base en introduisant dans chaque contrainte de type *(Q)* ou *(R)* une nouvelle variable dite « variable artificielle ».
- Le coefficient d'une variable artificielle est 1, et la colonne associée à cette variable est unitaire.

4.5. Méthode des deux phases

Objectif

L'objectif de la méthode des deux phases consiste à utiliser des variables artificielles pour construire une base initiale et chercher à faire sortir toutes les variables artificielles de la base en itérant l'algorithme du simplexe. L'obtention d'une base sans variables artificielles permet d'avoir une solution de base où les variables artificielles sont nulles. On obtiendra ainsi un sommet admissible du polyèdre.

Méthode

On substitue à la fonction économique *Z* une autre fonction *Z'* telle que *Z'* est la somme des variables artificielles. Le nouveau problème consiste à minimiser *Z'*. A l'optimum de ce nouveau PL, il y'a 2 cas:

- *Z'=0* : on obtient un sommet du polyèdre *D*, on reprend la solution du PL initial à partir de ce sommet.
- *Z'>0* : toute base admissible du nouveau PL comporte au moins une variable artificielle strictement positive, ce qui signifie que le PL d'origine est impossible (les contraintes sont contradictoires).

4.6. Méthode du «grand M»

La méthode du grand M consiste à combiner les deux phases: introduire les variables artificielles dans les contraintes puis dans la fonction économique d'origine. Chaque variable artificielle x_{ai} sera munie d'un coefficient $-M$ $(M \to +\infty)$.

L'objectif est de faire sortir les variables artificielles de la base en maximisant la nouvelle fonction économique $\hat{Z}$ telle que: $\tilde{Z} = Z - M.\sum_i x_{a_i}$.

A l'optimum de ce PL (avec $\hat{Z}$), il y'a deux cas :

1) Toutes les variables artificielles sont nulles: $\hat{Z}$ coïncide avec la fonction économique d'origine Z.
2) Il existe une variable artificielle en base et non nulle $(\hat{Z} \to -\infty)$. Dans ce cas le PL d'origine est impossible.

5. Dualité

5.1. Définition du programme dual

A chaque programme linéaire on peut associer un autre programme linéaire nommé «programme dual», dans ce cas le 1[er] PL est appelé «programme primal». Les propriétés des deux programmes primal et dual sont étroitement liées.

Forme standard de passage au dual

Considérons un PL sous la forme suivante :

$$P : \begin{cases} A.x \le b, x \ge 0 \\ [\max]\ Z = C.x \end{cases}$$

Cette forme est appelée la forme standard de passage au dual.

Définition et règles de passage au dual

On note y le vecteur-ligne $1 \times m$ (des variables duales). Le programme dual D du programme P est le suivant :

$$D : \begin{cases} y.A \ge C, y \ge 0 \\ [\min]\ Z' = y.b \end{cases}$$

Exemple 1.5

Soit le PL suivant :

$$\begin{cases} x_1 \le 1000 \\ x_2 \le 500 \\ x_3 \le 1500 \\ 3x_1 + 6x_2 + 2x_3 \le 6750 \\ x_1, x_2, x_3 \ge 0 \\ 4x_1 + 12x_2 + 3x_3 = Z\ [\max] \end{cases}$$

Le dual de ce PL:

$$\begin{cases} y_1 + 3y_4 \ge 4 \\ y_2 + 6y_4 \ge 12 \\ y_3 + 2y_4 \ge 3 \\ y_1, y_2, y_3, y_4 \ge 0 \\ 1000y_1 + 500y_2 + 1500y_3 + 6750y_4 = Z'\ [\min] \end{cases}$$

5.2. Liens entre le primal et le dual

Théorème (Dual du dual)

Le dual du dual d'un programme linéaire coïncide avec le primal.

Théorème (Cas possibles pour un primal-dual)

Pour deux PL en dualité (un primal et un dual), un et un seul des 3 cas suivants se trouve réalisé :

- *Cas 1:* Le primal et le dual ont des solutions optimales finies, respectivement x^* et y^*. A l'optimum, les valeurs des fonctions économiques du primal et du dual sont égales :
$$Z^* = Z'^* \quad (C.x^* = y.b^*)$$
- *Cas 2:* L'un des deux PL n'a pas de solution admissible. L'autre PL en admettant au moins une solution mais son optimum est rejeté à l'infini.
- *Cas 3:* Ni le dual ni le primal n'ont de solutions admissibles.

Illustration des cas

- *Cas 1:* Le problème de l'atelier de l'exemple 1.1.
- *Cas 2:* On vérifie que l'optimum de (P) est rejeté à l'infini, et le dual est impossible (son polyèdre est vide):

$$(P): \begin{cases} -x_1 + x_2 \leq 1 \\ -x_1 - x_2 \leq -2 \\ x_1, x_2 \geq 0 \\ 3x_1 + 3x_2 = Z \text{ [max]} \end{cases} \qquad (D): \begin{cases} -y_1 - y_2 \geq 3 \\ y_1 - y_2 \geq 3 \\ y_1, y_2 \geq 0 \\ y_1 - 2y_2 = Z' \text{ [min]} \end{cases}$$

- *Cas 3:* Ni (P) ni (D) n'ont de solutions admissibles.

$$(P): \begin{cases} x_1 - x_2 \leq 3 \\ -x_1 + x_2 \leq -4 \\ x_1, x_2 \geq 0 \\ 2x_1 + x_2 = Z \text{ [max]} \end{cases} \qquad (D): \begin{cases} y_1 - y_2 \geq 2 \\ -y_1 + y_2 \geq 1 \\ y_1, y_2 \geq 0 \\ 3y_1 - 4y_2 = Z' \text{ [min]} \end{cases}$$

5.3. Critère d'optimalité

Théorème (Inégalité entre Z et Z')

Supposons que le primal et le dual aient des solutions admissibles. Si x et y sont respectivement des solutions admissibles du primal et du dual alors: $Z(x) \leq Z'(y) \qquad (C.x \leq y.b)$

Théorème (Critère d'optimalité)

S'il existe une solution admissible $\tilde{x}$ du primal et une solution admissible $\tilde{y}$ du dual telles que : $Z(\tilde{x}) = Z'(\tilde{y})$ $(C.\tilde{x} = \tilde{y}.b)$ Alors $\tilde{x}$ est une solution optimale du primal et $\tilde{y}$ est une solution optimale du dual.

5.4. Correspondance entre l'optimum du primal et l'optimum du dual

L'optimum du primal et l'optimum du dual sont liés par les relations suivantes :

$$Z(x) = Z'(y)$$
$$y_i = -\Delta_{\bar{i}} \text{ et } x_{\bar{i}} = -\Delta'_i \quad \text{pour } i = 1,2,3,\ldots,m$$
$$y_{\bar{j}} = -\Delta_j \text{ et } x_j = -\Delta'_{\bar{j}} \quad \text{pour } j = 1,2,3,\ldots,n$$

Correspondance d'appartenance à la base : variables primales-variables duales:

- Si x_j est dans la base optimale du primal (respectivement hors base), alors $y_{\bar{j}}$ est hors base au dual (respectivement dans la base).
- Si y_i est dans la base optimale du dual (respectivement hors base), alors $x_{\bar{i}}$ est hors base au primal (respectivement dans la base).

Correspondance entre les coefficients des tableaux optimaux : Primal-Dual

En se limitant aux colonnes hors-base (les colonnes de base sont unitaires), on donne les équations de correspondance : $\alpha_{kl} = -\alpha'_{\bar{l}\,\bar{k}}$ et $\alpha'_{kl} = -\alpha_{\bar{l}\,\bar{k}}$

Remarque

Une manière pratique d'appliquer ces règles de correspondance est de condenser dans un tableau unique le tableau du primal et celui du dual.

5.5. Relations d'exclusion

Considérons deux PL en dualité :

Le Primal (P) : [A.x ≤ b, x ≥ 0, max Z=C.x]

Le Dual (D) : [y.A ≥ C, y ≥ 0, min Z'=y.b]

Théorème (Relations d'exclusion)

Soient $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$ et $\tilde{y}=(\tilde{y}_1,\tilde{y}_2,\ldots,\tilde{y}_m)$ respectivement des solutions admissibles du primal et du dual.

$\tilde{x}$ et $\tilde{y}$ sont des solutions optimales *ssi* :

$$\forall i = 1,2,\ldots,m : \tilde{y}_i.\tilde{x}_{\bar{i}} = 0 \text{ soit : } \tilde{y}_i.(b_i - \sum_{j=1}^{n} a_{ij}.\tilde{x}_j) = 0 \quad (1)$$

$$\forall j = 1,2,\ldots,n : \tilde{y}_{\bar{j}}.\tilde{x}_j = 0 \text{ soit : } (\sum_{i=1}^{m} \tilde{y}_i.a_{ij} - c_j).\tilde{x}_j = 0 \quad (2)$$

Les relations d'exclusions permettent de trouver l'optimum de dual si l'on connait l'optimum du primal et réciproquement.

Exemple 1.6

Soit le sous problème de l'exemple de l'atelier :

$$(P) : \begin{cases} x_1 \le 1000 \\ x_2 \le 500 \\ x_1 + 2x_2 \le 1750 \\ x_1, x_2 \ge 0 \\ \max\ Z = 4x_1 + 12x_2 \end{cases}$$

En appliquant le théorème des relations d'exclusion on a:

$$\begin{cases} y_1.(1000 - 750) = 0; y_2.(500 - 500) = 0; y_3.(1750 - 1750) = 0 \\ (y_1 + y_3 - 4).750 = 0; (y_2 + 2y_3 - 12).500 = 0 \end{cases}$$

$$\begin{cases} y_1^* = 0; y_2^* = 4; y_3^* = 4 \\ Z'^* = 9000 \text{ (car } Z'^* = Z^*) \end{cases}$$

D'où l'optimum de *(D)*.

Les relations d'exclusions permettent de tester si une solution admissible du primal (ou du dual) est optimale ou pas : il suffit de reporter les valeurs des variables du primal (respectivement du dual) dans le système des relations d'exclusion. Si on obtient une solution admissible du dual (respectivement du primal) alors la solution testée du primal (respectivement du dual) est optimale. Si on n'obtient pas une solution admissible du dual (respectivement du primal) alors la solution testée du primal (respectivement du dual) n'est pas optimale.

Exemple 1.7

Soit la solution admissible de l'exemple précédent : $x_1=850$, $x_2=450$.

On vérifie si cette solution est optimale :

On a : [y_1.(1000-850)=0 ; y_2.(500-450)=0 ;y_3.(1750-1750)=0]

Et par conséquent : $y_1=y_2=0$

On a aussi : [(y_3-4).850=0 ; (2y_3-12).450=0]

Il s'ensuit que : $y_3=4$ et $y_3=6$

Cette solution admissible *(850,450)* n'est pas optimale.

Remarque

- Les relations d'exclusion s'appliquent à des solutions admissibles de *(P)* et *(D)*, mais pas nécessairement de base.
- Si une contrainte du primal (respectivement du dual) n'est pas saturée $x_{\bar{i}} \neq 0$ (respectivement $y_{\bar{j}} \neq 0$) alors la variable duale (respectivement primale) associée à cette contrainte est nulle : $y_i=0$ (respectivement $x_j=0$).
- Si une variable du primal x_j (respectivement du dual y_i) est non nulle, alors la variable d'écart du dual $y_{\bar{j}}$ (respectivement du primal $x_{\bar{i}}$) est nulle.

6. Exercices

Exercice 1.1

Soit le programme suivant :

$$\begin{cases} x_1 \leq 1000 \\ x_2 \leq 500 \\ x_3 \leq 1500 \\ 3x_1 + 6x_2 + 2x_3 \leq 6750 \\ x_1, x_2, x_3 \geq 0 \\ [\max]Z = 4x_1 + 12x_2 + 3x_3 \end{cases}$$

Résoudre ce système en utilisant le procédé algébrique du simplexe.

Exercice 1.2

Soit le programme suivant :

$$\begin{cases} x_1 + x_4 = 1000 \\ x_2 + x_5 = 500 \\ x_3 + x_6 = 1500 \\ 3x_1 + 6x_2 + 2x_3 + x_7 = 6750 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \\ [\max]Z = 4x_1 + 12x_2 + 3x_3 \end{cases}$$

Résoudre ce système en utilisant la méthode des tableaux du simplexe.

Exercice 1.3

Soit un PL écrit en forme standard.

Montrer que si l'on connaît deux solutions différentes de ce système, alors toute combinaison linéaire convexe de ces 2 solutions est aussi une solution du système.

Exercice 1.4

Soit le programme : $[\max]\ Z = 6x_1 + 4x_2$

Sous les contraintes:

$$\begin{cases} -3x_1 + 2x_2 \leq 4 \\ 3x_1 + 2x_2 \leq 16 \\ x_1 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

1- Résoudre ce problème par la méthode géométrique.
2- Quels sont les points optimaux ?
3- Quelle est la particularité de ce programme ?
4- Résoudre ce problème par la méthode des tableaux du simplexe.

Exercice 1.5

Soit le programme : $[\max]\ Z = x_1 + x_2$

Sous les contraintes:

$$\begin{cases} -3x_1 + 2x_2 \leq 4 \\ 3x_1 + 2x_2 \leq 16 \\ x_1 \leq 3 \\ x_1 + 4x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

1- Résoudre ce problème par la méthode géométrique.
2- Résoudre ce problème par la méthode des tableaux
3- Donner un sommet du polyèdre des solutions admissibles associé à plusieurs bases optimales.
4- Quelles sont les particualrités de ce programme au niveau graphique et au niveau du tableau optimal ?
5- Quel est le rôle de la contrainte 3 dans le problème? justifier

Exercice 1.6

Une usine X produit 2 types de produits pour lesquels elle utilise 3 matières premières : A,B et C.
Le tableau ci-dessous résume les données du problème.

	A	B	C	Bénéfice
Produit 1	2,5 kg	125 g	17,5 kg	65
Produit 2	7,5 kg	125 g	10 kg	115
Quantités disponibles	240 kg	5 kg	595 kg	

Pour fabriquer une unité du produit 1, l'usine utilise *2,5 kg* de A , *125 g* de B et *17,5 kg* de C.
La fabriquation de cette unité lui rapporte alors un bénéfice de 65 dh.

1- Déterminer la fabriquation optimale de l'usine graphiquement. Vérifier par la méthode des tableaux.

Une usine Y concurrente demande à l'usine X de lui vendre 50% de son stock de la matière B (avant le lancement de la fabrication déterminée en 1).

2- À quel prix minimum X devera-t-elle lui vendre le Kg de la matière B ?

Exercice 1.7

On considère le système linéaire suivant :

$$\begin{cases} x_1 \leq 1000 \\ x_2 \leq 500 \\ x_1 + 2x_2 \leq 1750 \\ x_1, x_2 \geq 0 \end{cases}$$

1- Donner le polyèdre D des solutions admissibles de ce système.
2- Donner toutes les bases admissibles et non admissibles qu'on peut associer à ce système.
3- Pour chaque base admissible donner le sommet du polyèdre D correspondant.

Exercice 1.8

On considère le PL suivant :

$$\begin{cases} x_1 + x_2 - x_3 + x_4 = 1 \\ 2x_1 + 4x_3 + 2x_4 + x_5 = 7 \\ x_1 + 6x_2 + x_3 + 2x_5 = 19 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \\ x_1 + 3x_2 + 5x_3 + x_4 + 4x_5 = Z[\max] \end{cases}$$

1- Vérifier que (0,2,1,0,3) est une solution de base réalisable.
2- Ecrire sous forme matricielle les équations correspondantes pour la résolution du système par l'algorithme du simplexe.
3- Résoudre le problème par la méthode des tableaux.

Exercice 1.9

On considère le PL suivant :

$$\begin{cases} x_1 + 2x_2 + x_3 \leq 5 \\ 2x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 = 4 \\ x_1, x_2, x_3 \geq 0 \\ 4x_1 + 5x_2 + 3x_3 = Z \text{ [max]} \end{cases}$$

Résoudre ce PL en utilisant la méthode du grand M.

Exercice 1.10

Soit le PL suivant :

$$\begin{cases} x_1 + 2x_2 + x_3 \leq 5 \\ 2x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 = 4 \\ x_1, x_2, x_3 \geq 0 \\ 4x_1 + 5x_2 + 3x_3 = Z \text{ [max]} \end{cases}$$

Résoudre ce PL en utilisant la méthode des deux phases.

Exercice 1.11

Par la méthode des tableaux, résoudre le programme linéaire:

$$\begin{cases} 2x_1 + 2x_2 - x_3 \geq 2 \\ 3x_1 - 4x_2 \leq 3 \\ x_2 + 3x_3 \leq 5 \\ x_1, x_2, x_3 \geq 0 \\ 5x_1 - 2x_2 + 3x_3 = Z \text{ [max]} \end{cases}$$

On utilisera la méthode du grand M.

Exercice 1.12

Soit le PL suivant :

$$\begin{cases} 2x_1 + x_2 - 4x_3 \leq 10 \\ x_1 - 5x_2 + 6x_3 \geq 2 \\ 3x_1 - 2x_2 + 7x_3 = 5 \\ x_1, x_2, x_3 \geq 0 \\ \text{[min]}Z = -2x_1 + 3x_2 + 4x_3 \end{cases}$$

Donner le programme dual associé à ce PL.

Exercice 1.13

Soit P et D deux programmes linéaire en dualité. On note D_1 et D_2 respectivement les polyèdres des solutions admissibles de P et D.

Nous supposons que D_1 et D_2 ne sont pas vides. Montrer que :

$$\forall (x,y) \in D_1 \times D_2 : Z(x) \leq Z'(y)$$

Où Z et Z' sont respectivement les deux fonctions économiques du primal et du dual.

Exercice 1.14

On considère deux PL en dualité.

Soient $\tilde{x}$ et $\tilde{y}$ respectivement des solutions admissibles du primal et du dual.

Montrer que si $Z(\tilde{x}) = Z(\tilde{y})$ alors $\tilde{x}$ est une solution optimale du primal et $\tilde{y}$ est une solution optimale du dual.

Exercice 1.15

Soit le PL suivant :

$$\begin{cases} x_1 \leq 1000 \\ x_2 \leq 500 \\ x_3 \leq 1500 \\ 3x_1 + 6x_2 + 2x_3 \leq 6750 \\ x_1, x_2, x_3 \geq 0 \\ 4x_1 + 12x_2 + 3x_3 = Z\text{[max]} \end{cases}$$

1- Donner le dual de ce PL.

2- Résoudre par la méthode des tableaux du simplexe le problème dual.

3- Comparer le tableau optimal du dual avec celui du primal.

Exercice 1.16

Soit à maximiser la fonctionnelle linéaire : $5x_1 + 4x_2 + 3x_3 + x_4 = Z$

Sous les contraintes:

$$\begin{cases} x_1 + x_2 + x_3 + x_4 \leq 12 & (1) \\ 3x_1 + 2x_2 + 3x_3 + x_4 \leq 5 & (2) \\ 2x_1 + x_2 + x_3 + 4x_4 \leq 7 & (3) \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

1- Résoudre ce problème par l'algorithme du simplexe.

2- Donner , sans calcul, l'optimum du programme dual correspondant.

3- Montrer que la contrainte (1) du prorgamme primal est redondante.

4- Tenant compte de ce fait, simplifier le programme dual puis montrer que l'inéquation la plus contraignante du dual est la contrainte (2).

5- Résoudre graphiquement le programme dual.

Chapitre 2 : Généralités sur les graphes

1. Introduction à la théorie des graphes

1.1. Graphes orientés

Définition (Graphe)

Un graphe orienté $G = (X, U)$ est défini par la donnée de 2 ensembles X et U :

- Un ensemble X dont les éléments sont appelés sommets ou nœuds.
- Un ensemble U dont les éléments $u \in U$ sont appelés des arcs et constituent des couples ordonnés de sommets.

Remarque

- Le graphe G est un ensemble de sommets qui sont reliés entre eux par des arcs.
- Dans ce cours on ne considère que des graphes finis pour lesquels X et U sont des ensembles finis.
- L'ensemble U est un sous ensemble du produit cartésien X^2.

Exemple 2.1

Dans le graphe $G=(X,U)$ (fig 2.1) on a : $4 \in X$ et $(3,4) \in U$

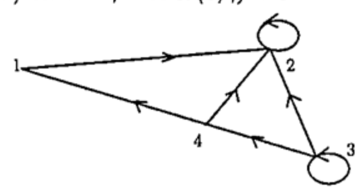


Figure 2.1

Exemple 2.2

Modélisation d'un programme (fig.2.2) par le graphe du flot de contrôle (fig.2.3):

```
double calcul (int x, int y)
{          int n=abs(y) ;
           double p=1.0 ;
           while(n !=0){
           p=p*x ;
           n=n-1 ;}
if (y<0)p=1.0/p;
return(p);
}
```

Figure 2.2

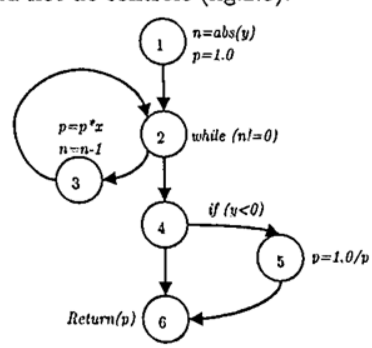


Figure 2.3

Définitions

Soit $G = (X,U)$ un graphe et $u=(i,j)$ un arc de G.

- Le sommet i est appelé l'extrémité initiale de l'arc $u=(i,j)$ et on note $i=I(u)$.
- Le sommet j est appelé l'extrémité terminale de l'arc $u=(i,j)$ et on note $j=T(u)$.
- L'arc u est une boucle si $i=j$.
- Le graphe G est sans boucles si U ne contient pas d'arcs de la forme (x,x).
- Le nombre de sommets de G est appelé ordre du graphe.
- Deux sommets x et y sont dits adjacents si il existe l'un des 2 arcs (x,y) ou (y,x) dans U.
- i et j sont dits incidents à l'arc $u=(i,j)$.
- L'arc $u= (i,j)$ est incident aux sommets i et j.
- Deux arcs sont adjacents s'ils sont incidents à un même sommet.

Remarque

- La disposition des points et la longueur des arêtes ou la forme n'ont aucune importance (fig.2.4):

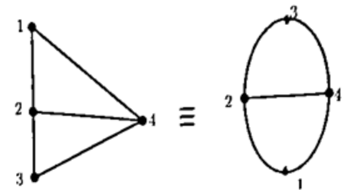


Figure 2.4

- Dans la représentation d'un graphe, deux arcs (ou arêtes) peuvent sembler, avoir une intersection en un point qui n'est pas un sommet. C'est le cas, par exemple, des arêtes e et f du graphe ci-dessous (fig.2.5) .Les arêtes e et f seront vues comme étant placées dans des plans différents et n'ayant donc aucun point commun:

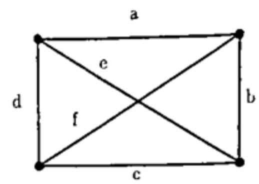


Figure 2.5

- La nature des objets représentés par les sommets n'a pas d'importance pour la manipulation des graphes.

Définition (Successeur-Prédécesseur d'un sommet)

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

- On dit que j est un successeur de i, s'il existe un arc ayant i comme extrémité initiale et j comme extrémité terminale.

Imp.

- L'ensemble des successeurs d'un sommet $i \in X$ est noté : $\Gamma_i (ou\ \Gamma(i))$.
- On dit que j est un prédécesseur de i, s'il existe un arc ayant j comme extrémité initiale et i comme extrémité terminale$((j,i) \in U)$.
- L'ensemble des prédécesseurs d'un sommet $i \in X$ est noté : $\Gamma_i^{-1} (ou\ \Gamma^{-1}(i))$.
- L'application Γ qui, associe tout élément $x \in X$ à une partie de X est appelée une application multivoque.

Remarque

- Un graphe peut être déterminé d'une façon unique par la donnée de l'ensemble X et l'application multivoque Γ de X à $P(X)$. On peut noter ce graphe $G=(X, \Gamma)$.

Définition (Arcs parallèles)

Soient e et v deux arcs distincts tels que $e=(x,y)$ et $v=(x,y)$.

On dit que e et v sont des arcs parallèles.

Un graphe sera dit simple s'il n'a pas d'arcs parallèles ni de boucles.

Remarque

(x,y) et (y,x) ne sont pas des arcs parallèles.

Définition (Degré, degré sortant et degré entrant d'un sommet)

Etant donné un graphe $G = (X,U)$ et un sommet x de G.

- Le cardinal de l'ensemble $\boxed{\{u \in U \ / \ I(u) = x\}}$ est appelé le demi degré extérieur (ou le degré sortant) du sommet x, et on le note $d_G^+(x)$: $\boxed{d_G^+(x) = \left|\{u \in U \ / \ I(u) = x\}\right|}$

- Le cardinal de l'ensemble $\{u \in U \ / \ T(u) = x\}$ est appelé le demi degré intérieur (ou le degré entrant) du sommet x, et on le note $d_G^-(x)$: $\boxed{d_G^-(x) = \left|\{u \in U \ / \ T(u) = x\}\right|.}$
- $d_G(x) = d_G^+(x) + d_G^-(x)$: est appelé le degré du sommet x : C'est le nombre d'arcs incidents au sommet x.

Remarque

- $d_G^+(x)$ est le nombre d'arcs dont x est l'extrémité initiale.
- $d_G^-(x)$ est le nombre d'arcs dont x est l'extrémité terminale.

Théorème (Somme des degrés)

Dans un graphe $G = (X,U)$, on a : $\sum_{x \in X} d_G(x) = 2.|U|$ Imp.

permet de resoudre 552 des exo. dans la theorie des graphes.

Preuve

Tout arc $e=(x,y)$ du graphe est compté exactement 2 fois dans la somme des degrés: c'est dans $d_G(x)$ et $d_G(y)$. Dont plus precise sera dans un exercice. >>>?

Corollaire

Soit $G = (X,U)$ un graphe. On a : $\sum_{x \in X} d_G^+(x) = \sum_{x \in X} d_G^-(x) = |U|$

Remarque

- Un sommet de degré entrant non nul et de degré sortant nul est appelé puits.
- Un sommet de degré entrant nul et de degré sortant non nul est appelé source.

Théorème

Dans un graphe, le nombre de sommets de degré impair est pair.

Preuve

Il suffit d'appliquer le théorème de la somme des degrés.

Définition (Sous graphe et graphe partiel)

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

Etant donnés deux ensembles A et V tels que : $A \subset X$ et $V \subset U$.

On appelle :

- Sous graphe engendré par A le graphe dont les sommets sont les éléments de A et dont les arcs sont les arcs de G ayant leurs extrémités dans A.
- Graphe partiel engendré par V le graphe ayant les mêmes sommets que G, et dont les arcs sont les arcs de V.
- Sous graphe partiel engendré par A et V le graphe dont les sommets sont les éléments de A et dont les arcs sont les arcs de V ayant leurs deux extrémités dans A.

Remarque

A tout graphe simple $G = (X,U)$, on peut faire correspondre une relation binaire R et une seule et réciproquement. Il suffit de poser : $\forall (x,y) \in X^2 : x \ R \ y \Leftrightarrow (x,y) \in U$

=> il s'agit d'une relation d'equivalence

Définition (Propriétés d'un graphe)

Un graphe sans arcs parallèles $G=(X,U)$ est dit :

Réflexif	si $(\forall i \in X)$: $(i,i) \in U$
Symétrique	si $(\forall (i,j) \in X^2)$: $(i,j) \in U \Rightarrow (j,i) \in U$
Antisymétrique	si $(\forall (i,j) \in X^2)$: $(i,j) \in U$ et $(j,i) \in U \Rightarrow j=i$
Fortement Antisymétrique	si $(\forall (i,j) \in X^2)$: $(i,j) \in U \Rightarrow (j,i) \notin U$
Transitif	si $(\forall (i,j,k) \in X^3)$: $(i,j) \in U, (j,k) \in U \Rightarrow (i,k) \in U$

1.2. Graphes non orientés

Définition (Graphe non orienté)

Un graphe non orienté $G=(X,U)$ est défini par la donnée de 2 ensembles X et U :

- Un ensemble X dont les éléments sont appelés sommets ou nœuds.
- Un ensemble U dont les éléments $u \in U$ sont appelés des arêtes et constituent des couples non ordonnés de sommets.

Remarque

- Le demi degré extérieur, intérieur sont des notions liées aux graphes orientés.
- On peut établir une équivalence entre un graphe orienté symétrique et un graphe non orienté en remplaçant les deux arcs : $a_1=(x,y)$ et $a_2=(y,x)$ par une arête $a=\{x,y\}$.

Définition (Graphe planaire)

Un graphe planaire est un graphe qui peut se représenter sur un plan sans qu'aucune arête n'en croise une autre.

Exemple 2.3

Le graphe K_4 est-il planaire (fig.2.6) ?

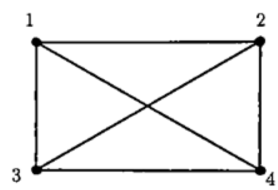


Figure 2.6

Définition (Graphe Complet)

Etant donné un graphe non orienté simple $G=(X,U)$.

G est dit complet si : $\forall \underset{(i \neq j)}{(i,j)} \in X^2 : \{i,j\} \in U$

Autrement dit tous les sommets du graphe doivent être liés deux à deux. G sera noté dans ce cas K_n où n est l'ordre de G (fig.2.7): $X=\{1,2,3,\dots,n\}$ et $U= P_2(X)$.

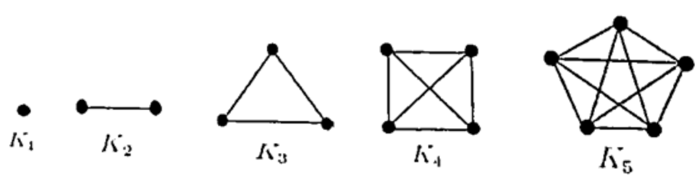


Figure 2.7

Définition (Clique)

On définit une clique d'un graphe G comme étant un sous graphe complet de G.

Définition (Graphe régulier)

Un graphe est dit régulier si tous ses sommets ont le même degré

Exemple 2.4:

Le graphe ci-dessous (fig.2.8) est 2-régulier:

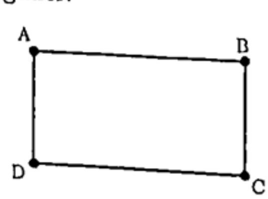


Figure 2.8

... calculer le diamètre d'un graphe quelconq, on ~~compte~~
emprunte toujour le chemin le plus court.

Définition (Graphe biparti)

Un graphe est dit biparti s'il existe une partition de son ensemble de sommets en deux sous-ensembles X_1 et X_2 telle que chaque arête ait une extrémité dans X_1 et l'autre dans X_2.

Définition (Graphe biparti-complet)

Le graphe biparti-complet $K_{p,q}$ est donné par :

$$X = \{x_1, x_2, \ldots, x_p, y_1, y_2, \ldots, y_q\} \text{ et } U = \{\{x_i, y_j\} \,/\, 1 \le i \le p \text{ et } 1 \le j \le q\}$$

Exemple 2.5 (Énigme des trois maisons)

«Un lotissement de trois maisons doit être équipé d'eau, de gaz et d'électricité. La règlementation interdit de croiser les canalisations pour des raisons de sécurité. Comment faut-il faire?»

La question devient : $K_{3,3}$ est-il un graphe planaire ?

Définition (Distance)

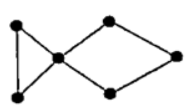
La distance entre deux sommets dans un graphe est définie par la longueur d'un plus court chemin entre ces deux sommets.

Définition (Diamètre)

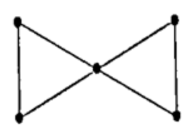
Le diamètre d'un graphe est la plus grande distance qui puisse exister entre deux de ses sommets.

Exemples de graphes

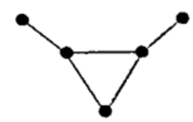
Graphe Cycle



Graphe Poisson



Graphe Papillon



Graphe Taureau

2. Chaînage et cheminement

2.1. Chaînes et cycles

Définition (Chaîne)

Soit $G=(X,U)$ un graphe et $x, y \in X$.

Une chaîne de longueur q joignant x à y dans G est une séquence $(u^1, u^2, \ldots, u^q)$ de q arcs de G telle que:

- Le premier arc u^1 de la séquence est incident à x par une de ses extrémités et au second arc u^2 de la séquence par son autre extrémité.
- Le dernier arc u^q de la séquence est incident à y par une de ses extrémités et à l'avant dernier arc u^{q-1} de la séquence par son autre extrémité.
- Chaque arc u^r de la séquence $(2 \le r \le q-1)$ est incident au précèdent u^{r-1} par une de ses extrémités et au suivant u^{r+1} par l'autre extrémité.

Définition (Chaîne Simple, Elémentaire)

Une chaîne est dite simple si la séquence d'arcs qui la constitue ne comporte pas plusieurs fois le même élément.

Une chaîne est dite élémentaire si en la parcourant on ne rencontre pas deux fois le même sommet.

Remarque

Une chaîne élémentaire est simple.

Définition (Cycle)

Un cycle est une chaîne simple dont les extrémités sont confondues.

Un cycle est dit élémentaire si en le parcourant on ne rencontre pas deux fois le même sommet (sauf le sommet choisi comme origine du parcours).

Remarque

Un cycle élémentaire est un cycle minimal (au sens de l'inclusion) c'est-à-dire ne contenant strictement aucun autre cycle.

2.2. Chemins et circuits

Définition (Chemin)

Soit $G = (X, U)$ un graphe et $x, y \in X$.

Un chemin de longueur q de x à y dans G est une séquence $(u^1, u^2, \ldots, u^q)$ de q arcs telle que:

- L'extrémité initiale du premier arc de la séquence est x.
- L'extrémité terminale du dernier arc de la séquence est y.
- L'extrémité initiale de chaque arc de la séquence (sauf le premier) coïncide avec l'extrémité terminale de l'arc précèdent.

Remarque

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

Définition (Chemin Simple, Elémentaire)

Un chemin est simple si la séquence d'arcs qui le constitue ne comporte pas plusieurs fois le même élément.

Un chemin est élémentaire si en le parcourant on ne rencontre pas deux fois le même sommet.

Remarque

Un chemin élémentaire est simple.

Définition (Circuit)

Un circuit est un chemin simple dont les deux extrémités sont confondues.

Un circuit est dit élémentaire si en le parcourant on ne rencontre pas deux fois le même sommet (sauf le sommet choisi comme origine du parcours).

Remarque

Si x_i et x_j sont 2 sommets d'un même circuit alors il existe 2 chemins de x_i à x_j et de x_j à x_i.

Exemple 2.6

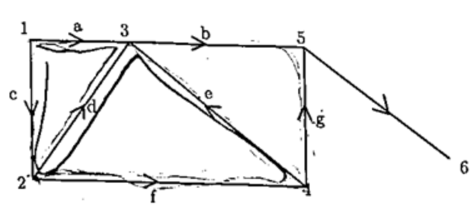


Figure 2.9

(a,e,g,b,e,f) est une chaîne non simple liant 1 et 2.

(c,d,e,f) chaîne simple non élémentaire liant 1 et 2.

(a,b,g,f) chaîne élémentaire liant 1 et 2.

(a,b,g,e,d,c) est un cycle non élémentaire.

(b,g,e) est un cycle élémentaire.

(a,d,f,e,d,c) n'est pas un cycle.

3. Modélisation d'un graphe

Un graphe peut être représenté en utilisant trois méthodes différentes :

- Matrice d'adjacence.
- Matrice d'incidence.
- Listes d'adjacence.

3.1. Matrice d'adjacence

La matrice d'adjacence est une matrice carrée à valeurs booléennes qui modélise la relation entre les sommets d'un graphe.

Définition (Matrice d'adjacence d'un graphe non orienté)

La matrice d'adjacence $X=(x_{ij})_{i,j}$ d'un graphe G d'ordre n non orienté et sans arêtes parallèles est une matrice carrée $(n \times n)$ dont chaque rangée et chaque colonne correspond à un sommet et dont les éléments sont définis comme suit:

$$\forall (i,j) \in X^2 : x_{ij} = \begin{cases} 1 \text{ s'il existe une arête entre i et j} \\ 0 \text{ autrement} \end{cases}$$

Remarque

- La présence de 1 sur la diagonale indique une boucle.
- La matrice est symétrique ($x_{ij} = x_{ji}$).
- La matrice contient $2.(m\text{-}b)+b$ cases non nulles (m est le nombre d'arêtes du graphe et b est le nombre de boucles de ce graphe).

Exemple 2.7

On détermine pour le graphe de la figure 2.10 la matrice d'adjacence X ainsi que quelques puissances successives de X :

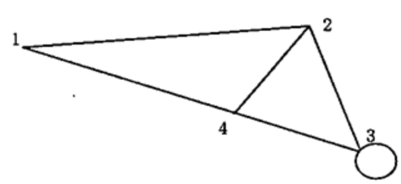


Figure 2.10

On a : d (3) = 4

$$X = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad X^2 = \begin{pmatrix} 2 & 1 & 2 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 3 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} \qquad X^3 = \begin{pmatrix} 2 & 5 & 4 & 5 \\ 5 & 5 & 7 & 6 \\ 4 & 7 & 7 & 7 \\ 5 & 6 & 7 & 5 \end{pmatrix}$$

Donner les chemins allant du sommet *1* au sommet *4* de longueur *3*. Comparer le nombre de ces chemins avec l'élément $x^{(3)}_{1,4}$ de la matrice X^3.

Définition (Matrice d'adjacence d'un graphe orienté)

La matrice d'adjacence $X=(x_{ij})_{i,j}$ d'un graphe G d'ordre n orienté sans arcs parallèles est une matrice carrée $(n \times n)$ dont chaque ligne et chaque colonne correspond à un sommet et dont les éléments sont définis comme suit:

$$\forall (i,j) \in X^2 : x_{ij} = \begin{cases} 1 \text{ s'il existe un arc (i,j)} \\ 0 \text{ autrement} \end{cases}$$

Remarque:

- La matrice est non symétrique.
- La matrice contient m cases non nulles (m est le nombre d'arcs du graphe).

Exemple 2.8

On détermine pour le graphe de la figure 2.11 la matrice d'adjacence X ainsi que quelques puissances successives de X :

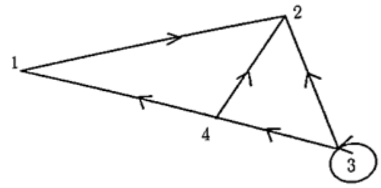


Figure 2.11

$$X = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \qquad X^2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

3.2. Matrice d'incidence

La matrice d'incidence est une matrice à trois valeurs 1,-1 et 0 qui représentent la relation Sommet-Arc dans un graphe sans boucles.

Définition (Matrice d'incidence d'un graphe orienté)

La matrice d'incidence $A=(a_{ij})_{ij}$ aux arcs d'un graphe orienté G sans boucles de n sommets et de m arcs est une matrice $(n \times m)$ dont chaque ligne correspond à un sommet et chaque colonne à un arc et dont les éléments sont définis comme suit:

$$(\forall i \in X),(\forall j \in U) : a_{ij} = \begin{cases} 1 \text{ si l'arc j sort du sommet i : } i = I(j) \\ -1 \text{ si l'arc j arrive au sommet i : } i = T(j) \\ 0 \text{ qutrement (si i n'est pas une extremité de j)} \end{cases}$$

Exemple 2.9

On détermine pour le graphe de la figure 2.12 la matrice d'incidence A:

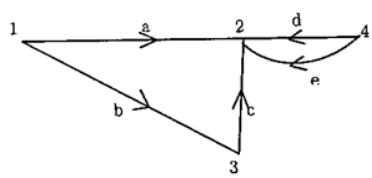


Figure 2.12

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

La matrice est Compos de Liens entre les ≠ts sommets du graphe

Remarque

- La présence d'une boucle j au niveau d'un sommet $i \Rightarrow a_{ij} = \pm 1$. Ce qui rend les boucles interdites.
- La matrice contient $2.m$ cases non nulles.

Définition (Matrice d'incidence d'un graphe non orienté)

Soit $G=(X,U)$ un graphe non orienté sans boucles tel que :

$$X= \{x_1, x_2, \dots, x_n\} \text{ et } U=\{ u_1, u_2, \dots, u_m \}.$$

La matrice d'incidence $A=(a_{ij})_{ij}$ aux arêtes est une matrice $(n \times m)$ dont les éléments sont définis comme suit :

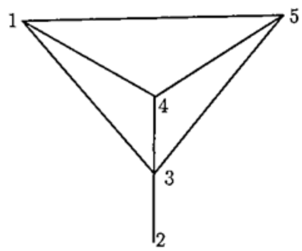
$$\forall(i,j) \in \{1,2,\dots,n\} \times \{1,2,\dots,m\} : a_{ij} = \begin{cases} 1 \text{ si } x_i \text{ est une extrémité de l'arc } u_j \\ 0 \text{ autrement} \end{cases}$$

3.3. Listes d'adjacence

Les listes d'adjacence permettent de représenter un graphe en donnant pour chaque sommet la liste de ses sommets successeurs (ou prédécesseurs). Si le graphe est non orienté on donne pour chaque sommet la liste de ses sommets adjacents.

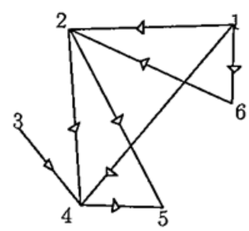
Exemple 2.10

On représente respectivement les listes d'adjacence pour un graphe non orienté (fig.2.13) et les sommets successeurs pour un graphe orienté (fig.2.14):



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

Figure 2.13



1 : 2, 4, 6
2 : 4,5
3 : 4
4 : 5
5 :
6 : 2

Figure 2.14

Ce dernier graphe (fig.2.14) peut être implémenté par une liste chainée (fig.2.15):

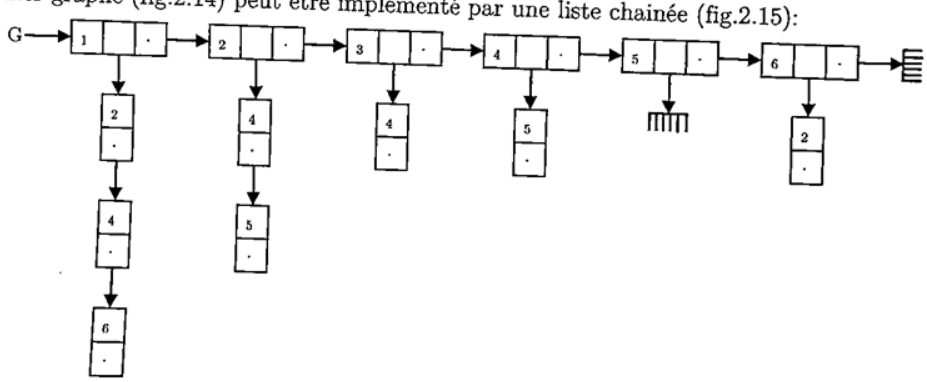


Figure 2.15

4. Connexité et forte connexité d'un graphe

La notion de connexité d'un graphe est le fait que tout sommet du graphe peut être atteint en suivant une suite d'arcs (chaine). En pratique il faut vérifier pour un réseau quelconque (réseau informatique, réseau routier, réseau de communication téléphonique...) si cette configuration choisie permet d'avoir un lien de n'importe quel point à n'importe quel autre.

4.1. Connexité d'un graphe

Définition (Graphe connexe)

Un graphe G est connexe s'il existe au moins une chaine entre les deux éléments de toute paire de sommets.

Définition (Relation de connexité)

La connexité est une relation binaire notée C sur l'ensemble des sommets X d'un graphe $G_{(X,U)}$:

$$\forall(x,y) \in X^2 : x \ C \ y \text{ Si:} \begin{cases} x = y \\ \text{ou} \\ \text{il existe une chaîne joignant } x \text{ et } y \end{cases}$$

Remarque

- La relation de connexité est une relation d'équivalence sur X.
- Les classes d'équivalence sur X induites par la relation C forment une partition de X.
- Les classes d'équivalences sont appelées les composantes connexes de G.
- Si x_i et x_j sont deux sommets du graphe alors $\overline{x}_i = \overline{x}_j$ ou $\overline{x}_i \cap \overline{x}_j = \varnothing$.
- Le nombre de classes d'équivalence distinctes est appelé le nombre de connexité du graphe.
- Un graphe est connexe ssi son nombre de connexité est 1.
- Chaque composante connexe est un graphe connexe.
- Un sous graphe engendré par une composante connexes est appelé aussi composante connexe.

Définition (Point d'articulation-Isthme)

- On appelle point d'articulation d'un graphe un sommet dont la suppression augmente le nombre de composantes connexes.
- On appelle isthme une arête dont la suppression augmente le nombre de composantes connexes.

Exemple 2.11

Ces deux graphes sont connexes (fig.2.16)

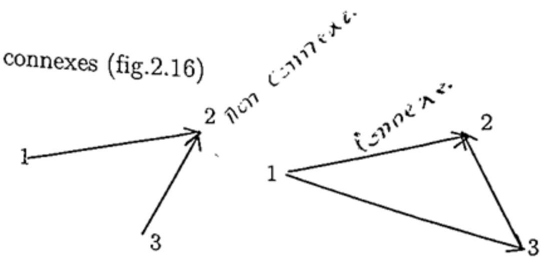


Figure 2.16

4.2. Forte connexité d'un graphe

Définition (Graphe fortement connexe)

Un graphe orienté est dit fortement connexe s'il existe un chemin joignant les deux éléments de toute paire de sommets.

Définition (Relation de forte connexité)

La forte connexité est une relation binaire notée $\tilde{C}$ sur l'ensemble des sommets X d'un graphe orienté $G = (X,U)$:

$$\forall(x,y) \in X^2 : x \ \tilde{C} \ y \text{ Si :} \begin{cases} x = y \\ \text{ou} \\ \text{il existe dans } G \text{ un chemin de } x \text{ à } y \text{ et un chemin de } y \text{ à } x \end{cases}$$

Remarque

- La relation de forte connexité est une relation d'équivalence.
- Les classes d'équivalence sur X induites par la relation $\tilde{C}$ forment une partition de X.
- Les classes d'équivalences sont appelées les composantes fortement connexes de G.
- Si x_i et x_j sont deux sommets du graphe alors $\overline{\overline{x}}_i = \overline{\overline{x}}_j$ ou $\overline{\overline{x}}_i \cap \overline{\overline{x}}_j = \varnothing$.
- Le nombre de classes d'équivalence distinctes est appelé le nombre de forte connexité du graphe.
- Un graphe est fortement connexe ssi son nombre de forte connexité est 1.

- Chaque composante fortement connexe est un graphe fortement connexe.
- Un sous. graphe engendré par une composante fortement connexes est appelé aussi composante fortement connexe.

4.3. Graphe réduit d'un graphe

Définition (Graphe réduit)

Etant donnée un graphe $G = (X,U)$.

On appelle graphe réduit de G le graphe $\hat{G} = (\hat{X},\hat{U})$ tel que :

- Les éléments de $\hat{X}$ sont les composantes fortement connexe de G.
- Un arc (C_i, C_j) appartient à $\hat{U}$ s'il existe au moins dans le graphe G un arc entre un sommet de la composante C_i et un sommet de la composante C_j.

Exemple 2.12

Le graphe G de la figure 2.17 est connexe mais il n'est pas fortement connexe :

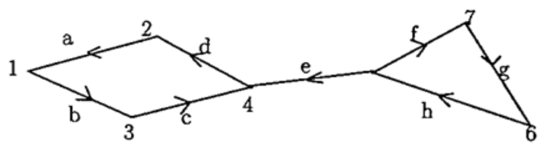
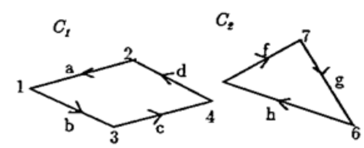


Figure 2.17

Ce graphe a 2 composantes fortement connexes C_1 et C_2:



Le graphe réduit de G : $C_1 \longleftarrow C_2$

4.4. Algorithmes de connexité ← (On va pas le faire !)

Recherche d'une composante fortement connexe

Soit $G_{(X,U)}$ un graphe et soit $S \in X$.

Pour déterminer la composante fortement connexe qui contient S il suffit de :

- Déterminer l'ensemble des nœuds accessibles à partir de $S \rightsquigarrow X_1$.
- Déterminer l'ensemble des nœuds qui peuvent atteindre $S \rightsquigarrow X_2$.
- $CFC(S) = X_1 \cap X_2$.

On utilise les fonctions suivantes :

- *X= CreateSet( )* : crée l'ensemble vide X.
- *EmptySet(X)* : retourne un booléen qui indique si l'ensemble X est vide.
- *Insert(x,X)* : permet l'ajout d'un élément x dans l'ensemble X.
- *x= Delete(X)* : permet le retrait d'un élément de X.
- *Delete(x,X)* : permet le retrait de l'élément x de X.
- *Intersection(X, Y)* : permet de retourner l'intersection des ensembles X et Y.

Algorithme

L'algorithme de la figure 2.18 permet de déterminer la composante fortement connexe engendrée par un sommet quelconque:

```
G(X,U) :Graphe //Données d'entrées
n : Entier // Ordre du graphe
S: Sommet
Début
   X1,X2,C : Ensemble de sommets
   Traité : Tableau de booléens
   x,y,x1,x2 : sommets
   Allouer n cases pour le tableau Traité
   Pour tout sommets x dans X   Traité[x] = False;
   Fin Pour
        C= CreateSet( ); X1= CreateSet( ); Insert( S,X1);
   Pour tout sommet x dans X1
      Si ( Non(Traité[x]) Alors  Traité[x]= True Fin Si
      Pour tout arc (x,y) dans U
         Si ( Non(Traité[y])  Alors   Insert(y,X1); Fin Si
      Fin Pour
   Fin Pour
   X2= CreateSet( ); Insert( S,X2);
   Pour tout sommet x dans X2
      Si ( Non(Traité[x]) Alors Traité[x]= True; Fin Si
      Pour tout arc (y,x) dans U
         Si ( Non(Traité[y]) Alors  Insert(y,X2); Fin Si ;
      Fin Pour
   Fin Pour
   C=Intersection(X1,X2);
   Retourner C
Fin
```

Figure 2.18

Recherche de toutes les composantes fortement connexes

Soit $G_{(X,U)}$ un graphe.

Pour déterminer toutes les composantes fortement connexes il suffit de :

- Choisir un sommet arbitraire et déterminer sa composante fortement connexe X_1.
- Puis on choisit un sommet $\notin X_1$ et on cherche sa CFC X_2.
- ⋮
- Jusqu'à ce que tous les sommets font partie d'une CFC.

Algorithme

L'algorithme de la figure 2.19 permet de déterminer toutes les composantes fortement connexes pour un graphe quelconque:

```
G(X,U) : Graphe
n : Entier
Début
   C : Tableau d'ensembles
   X́ : Ensemble de sommets traités
   i : Entier ;
   x :Sommet;
   X́ = CreateSet ( ) ;
   Pour tout sommet  x de X
      Insert(x, X́ ) ;
   Fin Pour
   Allouer n cases pour le tableau C
   i=0 ;
   Tant que (Non(EmptySet(X́))
      x= Delete(X́); C[i]=CFC (G,n,x);
      Pour tout sommet x dans C[i]
         Delete (x,X́);
      Fin Pour
      i= i+1 ;
   Fin Tant que
   Retourner C
Fin
```

Figure 2.19

5. Coloration des graphes

5.1. Introduction

Le problème de coloration des graphes est un problème classique de la théorie des graphes qui permet de modéliser plusieurs situations pratiques rencontrées dans la vie quotidienne:

- Affecter des fréquences différentes à des cellules voisines dans un réseau de téléphone mobile.
- Organiser des examens suivant les matières que doit passer chaque étudiant en mettant en parallèle plusieurs épreuves.
- Optimiser l'utilisation des machines de travail en mettant en parallèle des fabrications utilisant plusieurs machines.
- Faire cohabiter des personnes ou des animaux en tenant compte de leur incompatibilité.

Dans ce paragraphe on suppose que $G = (X,U)$ est un graphe non orienté et simple.

5.2. Coloration des sommets d'un graphe

Définition (Stable de X)

S est un stable de X si $\begin{cases} S \subset X (S \neq \varnothing) \\ \forall \underset{(x \neq y)}{x, y} \in S \text{ : Le sommet } x \text{ n'est pas adjacent au sommet } y \end{cases}$

$\forall a \in X : \{a\}$ est un stable de X.

Définition (Nombre de stabilité)

On appelle nombre de stabilité de G, et on note $\alpha(G)$ le cardinal de la plus grande partie stable de X : $|Max\{S \in P(X) \ / \ S \text{ stable de } X\}|$

Définition (Coloration des sommets)

La coloration des sommets d'un graphe consiste à affecter une couleur à tous les sommets de ce graphe, de telle sorte que deux sommets adjacents ne portent pas la même couleur.

Une coloration avec k couleurs est donc une partition de l'ensemble des sommets en k parties stables.

Le nombre chromatique, noté $\chi_{(G)}$ du graphe G est le plus petit entier k pour lequel il existe une partition de X en k sous-ensembles stables.

Exemple 2.13

Une partition en stables des sommets de cet arbre (fig.2.20) est la suivante : $\{a, c, d, e\}, \{b, f\}$

Donc $\chi_{(G)} \leq 2$

Par exemple $\{b, c\}$ constitue une clique de G donc $\chi_{(G)} \geq 2$ et par conséquent $\chi_{(G)} = 2$

Cela signifie que 2 couleurs C_1 et C_2 sont suffisantes pour colorer les sommets de ce graphe (fig.2.20).

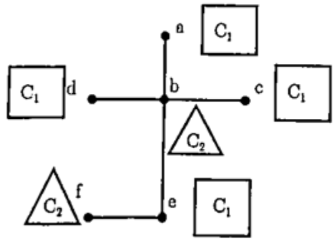


Figure 2.20

Théorème

Soit $G=(X,U)$ un graphe simple d'ordre n.

On a : $\chi_{(G)} + \alpha_{(G)} \leq n+1$

5.3. Coloration des arêtes d'un graphe

Définition (Coloration des arêtes)

Une coloration des arêtes d'un graphe G est une affectation de couleurs aux arêtes telle que les arêtes ayant une extrémité en commun sont de couleurs différentes. On cherche généralement à déterminer une coloration utilisant aussi peu de couleurs que possible. Le plus petit nombre de couleurs nécessaires pour colorer les arêtes d'un graphe G s'appelle : l'indice chromatique de G et noté $q(G)$.

Théorème de Vizing

On a : $\Delta(G) \leq q(\mathrm{G}) \leq \Delta(\mathrm{G})+1$ Où $\Delta(G)$ est le degré maximal de G.

Exemple 2.14

Pour le graphe ci-dessous (fig.2.21), on a : $\Delta(\mathrm{G}) = 4$

Par conséquent : $4 \leq q(G) \leq 5$

Une coloration des arêtes avec 4 couleurs est possible: $q(\mathrm{G}) = 4$

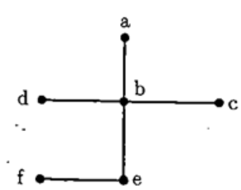


Figure 2.21

6. Exercices

Exercice 2.1

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

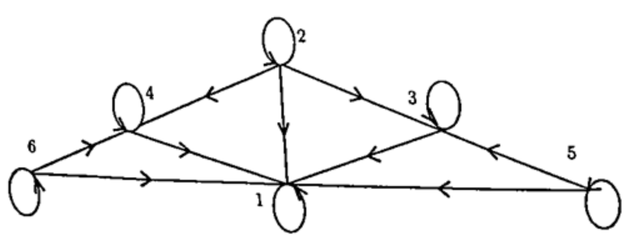
Montrer que :

G est symétrique ssi $\left[(\forall x \in X) : \Gamma(x) = \Gamma^{-1}(x)\right]$

Si G est fortement antisymétrique alors G est irréflexif.

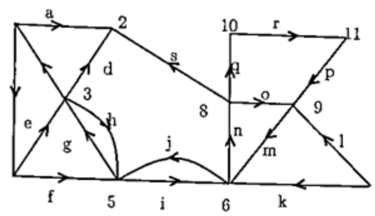
Exercice 2.2

Quelles sont les caractéristiques de la relation binaire associée au graphe suivant :



Exercice 2.3

Soit G le graphe suivant:



1- Donner les caractéristiques des chaînes suivantes :
(r,p,m,j,g,h,i,n,q,r), (d,e,f,h,d), (m,o,s,a,c,h,j) (h,i,n,o,m,j,g), (m,p,r,q,n,i,j)

2- Donner un chemin simple puis un chemin élémentaire maximal de G.

3- Déterminer les CFC et le graphe réduit G_r de G.

Exercice 2.4

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

Montrer que : $\sum_{x \in X} d_G^+(x) = \sum_{x \in X} d_G^-(x) = |U|$

Exercice 2.5

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

Montrer que le nombre de sommets de G de degrés impairs est pair.

Exercice 2.6

On note E_n l'ensemble des graphes non orientés n'ayant pas d'arêtes parallèles d'ordre n$(n \geq 2)$.

Donner le nombre de graphes de E_n.

Montrer que : $$\underset{G_{(X,U)} \in E_n}{Max} |U| = \frac{n(n+1)}{2}$$

Exercice 2.7

On considère un graphe non orienté simple et sans boucles $G = (X,U)$ dont l'ordre est n.

Montrer qu'il existe au moins deux sommets ayant le même degré.

Exercice 2.8

Donner le graphe $G = (X,U)$ dont la matrice d'incidence est la suivante :

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

Exercice 2.9

Soit la matrice d'incidence représentant le graphe G :

$$\begin{array}{c c} & \begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} +1 & & -1 & +1 & & \\ & -1 & & & & \\ -1 & & & & +1 & +1 \\ & & +1 & -1 & -1 & \\ & +1 & & & & -1 \end{pmatrix} \end{array}$$

1- Quel est le degré entrant (sortant) et le degré de chacun des sommets ?

2- Donner la matrice d'adjacence associée à G. Dessiner le graphe G.

Exercice 2.10

On considère le graphe $G=(X,U)$ d'ordre n dont la matrice d'adjacence U est donnée par:

$$U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

Chaque ligne u_k de la matrice est donnée par sa représentation décimale :

$u_1 = 2^{n-2}+1$

$u_n = 2u_1$

$u_k = 5.2^{n-k-1}, k \in \{2,3,\dots,(n-1)\}$

1- Donner la conversion binaire de chaque ligne.

2- Donner les éléments caractéristiques du graphe G.

Exercice 2.11

Calculer le nombre d'arêtes du graphe K_n en utilisant deux méthodes différentes.

Exercice 2.12

Etudier le graphisme (l'existence d'un graphe simple sans boucles $G(X,U)$) des suites suivantes:

u : 1,1,2,3,3 , v : 1,1,3,3 , x : 2,2,3,3 , y : 1,1,1,1,2,4 , z : 1,1,1,2,3,5 , t : 1,1,1,1,3,4,5

$a : \underbrace{(n,n,\dots,n)}_{(n+1)\ fois}$ $n \geq 1$ $b : \underbrace{(n^2,n^2,\dots,n^2)}_{n^2\ fois}$ $n \geq 2$

Exercice 2.13

Etant donnée la matrice carrée X=(x_{ij}) d'ordre n suivante : $\forall$ i,j : $x_{ij}=1$

1- Pourquoi X ne peut pas être une matrice d'incidence?

On suppose que X est une matrice d'adjacence :

1- Quelles sont les propriétés du graphe G correspondant à X ?

2- Quels sont les chemins de 1 à n de longueur 2 ?

3- Donner les éléments caractéristiques de G.

Exercice 2.14

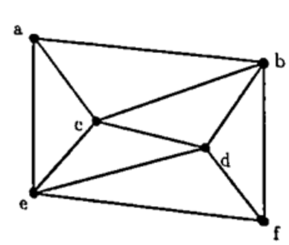
Donner la matrice d'adjacence du graphe C_{10}.

Exercice 2.15

Proposer une coloration (des sommets et des arêtes) pour les graphes suivants : le graphe papillon, le graphe poisson et le graphe taureau. En déduire le nombre et l'indice chromatique de ces graphes.

Exercice 2.16

Soit $G=(X,U)$ le graphe suivant:



1- Proposer une coloration pour les sommets de G.

2- Quel est le nombre chromatique de G ?

Exercice 2.17

Soit $G=(X,U)$ un graphe simple sans boucle d'ordre n. Montrer que :

$$\chi_{(G)}+\alpha_{(G)}\leq n+1$$

Exercice 2.18

Montrer par récurrence que le nombre de graphes non orientés d'ordre $n(n\geq 2)$ et n'ayant pas d'arêtes parallèles est: $2^{n(n+1)/2}$

Chapitre 3 : Arbres et Arborescences

1. Généralités sur les arbres

1.1 Arbres

Définition (Arbre)

Un arbre est un graphe connexe et sans cycles.

Remarque

- Un arbre est un graphe simple.
- L'orientation de l'arbre est sans importance.

Exemple 3.1

La figure 3.1 représente un arbre d'ordre 14

Figure 3.1

Théorème

Soit $G_{(X,U)}$ un graphe d'ordre n :

Si le graphe G est connexe alors $|U| \geq n-1$.

Si le graphe G est sans cycle alors $|U| \leq n-1$.

*) On peut extraire d'un graphe connexe un arbre

* Un arbre a au moin 2 fe

1 — 2

Remarque

Un arbre de n sommets comporte *(n-1)* arcs.

Théorème (Caractérisation des arbres)

Soit $G=(X,U)$ un graphe de $n \geq 2$ sommets.

Les propriétés suivantes sont équivalentes et caractérisent un arbre :

- G est connexe et sans cycles.
- G est sans cycle et comporte *(n-1)* arcs.
- G est connexe et comporte *(n-1)* arcs.
- G est sans cycle et maximal au sens des arcs pour cette propriété.
- G est connexe et minimal au sens des arcs pour cette propriété.
- Tout couple de sommets de G est relié par une chaîne élémentaire unique.

Remarque

Tout graphe connexe $G=(X,U)$ possède un graphe partiel qui est un arbre.

1.2. Forêts et Cycles fondamentaux

Définition (Feuille)

On appelle feuille d'un arbre (ou sommet pendant) tout sommet de degré 1.

Théorème

Un arbre $T=(X,U)$ de $n \geq 2$ sommets comporte au moins 2 feuilles. => 1 — 2 est l'arbre.

Définition(Forêt)

On appelle forêt tout graphe sans cycles.

Remarque

- Les composantes fortement connexes d'une forêt sont des arbres.
- L'orientation de la forêt est sans importance.

- Une forêt d'ordre n et à p composantes connexes possède *(n-p)* arcs.

Théorème de Cayley

Le nombre d'arbres T_n que l'on peut construire avec $n \geq 2$ sommets numérotés est égale à n^{n-2}.

Théorème (Cycle généré par une arête)

Soit $T_{(X,U)}$ un arbre, si on ajoute à T une arête u, le graphe $T' = (X, U \cup \{u\})$ contient un cycle et un seul.

Définition (Cycle fondamental)

L'unique cycle obtenu en ajoutant une arête à un arbre partiel T d'un graphe connexe G est appelé cycle fondamental relativement à T.

Remarque

Dans un graphe connexe G de n sommets et m arêtes, il existe par rapport à un arbre partiel T de G *(m-n+1)* cycles fondamentaux.

Exemple 3.2

Le graphe G est connexe et $T=\{v_1,v_2,v_3,v_4,v_5,v_6\}$ est un arbre partiel de G (fig.3.2):

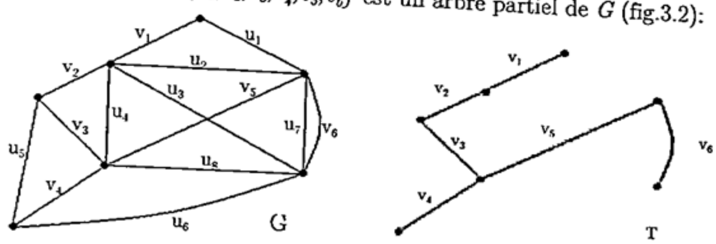


Figure 3.2

Si on ajoute u_2 à l'arbre T on crée le cycle fondamental $\{v_2,v_3,v_5,u_2\}$.

Si on ajoute $\{u_1,u_2\}$ à l'arbre T on crée 3 cycles.

2. Codage de Prüfer

Le codage de *Prüfer* est une représentation minimale d'un arbre de taille n $(n \geq 3)$.

On suppose que les sommets sont numérotés de *1* à *n*. Le code de *Prüfer* est une suite de *(n-2)* termes choisis dans l'ensemble *{1,2,…,n}*.

On pose: *T=(X,U)* où *X={1,2,…,n}*

2.1. Algorithme de codage

L'algorithme de codage de *Prüfer* permet de placer dans l'ensemble S les termes de la suite de codage de l'arbre T (fig.3.3):

```
S:=∅
Tant que (l'arbre courant contient plus que 2 sommets) Faire
    - Trouver la feuille qui a le numéro minimal
    - Mettre dans S le père de cette feuille dans l'arbre
      courant
    - Supprimer de l'arbre courant l'arête incidente à cette
      feuille
Fin Tant que
```

Figure 3.3

Exemple 3.3

La figure 3.4 représente une exécution de l'algorithme de codage de *Prüfer*:

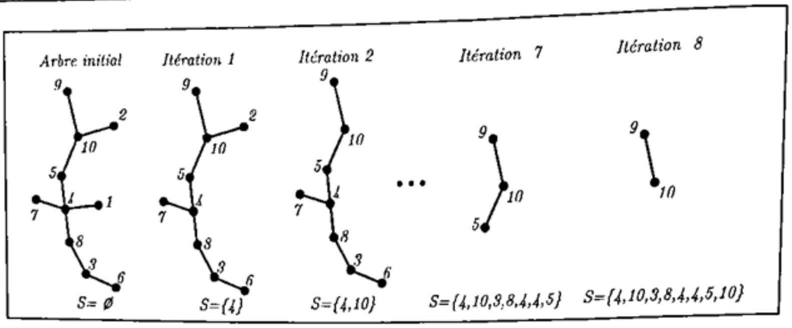


Figure 3.4

2.2. Algorithme de décodage

On dispose d'une suite S de N termes en provenance de l'ensemble $\{1,\ldots,(N+2)\}$.

On pose $I=\{1,2,\ldots,(N+2)\}$.

L'algorithme de décodage de *Prüfer* permet de tracer dans un nuage de sommets les arêtes de l'arbre T (fig.3.5):

```
Tant que (S≠Ø Et ∃ plus que 2 éléments dans I) Faire
    - Trouver le minimum min de I tel que min ∉S
    - Relier par une arête le sommet min avec le premier élément de S
    - Supprimer l'élément min de I et le premier élément de S
Fin Tant que
    - Relier par une arête les 2 sommets représentés par les 2 éléments de I
```

Figure 3.5

Exemple 3.4

La figure 3.6 représente une exécution de l'algorithme de décodage de *Prüfer*:

Le min de I qui n'$\boxed{\notin}$ à S

dans procédure du l'algo

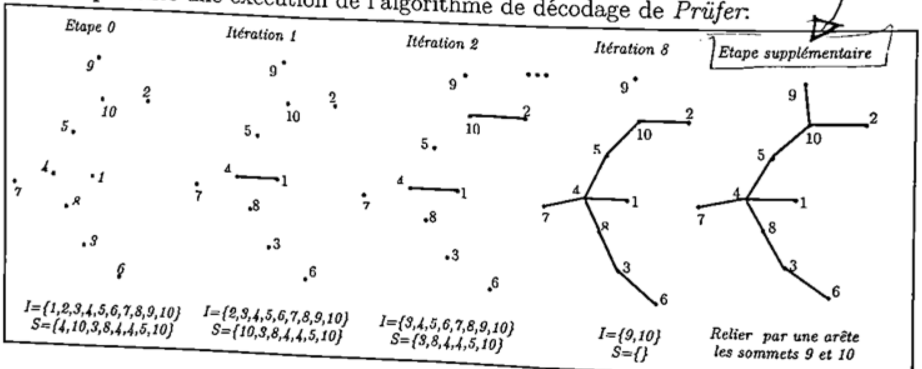


Figure 3.6

Fin le 5/1/2021

3. Problème de l'arbre couvrant de poids minimum

3.1. Position du problème

Définition (Arbre couvrant)

Un arbre couvrant est un graphe partiel d'un graphe connexe qui est aussi un arbre.

Définition du problème

Soit $G=(X,U)$ un graphe connexe.

Soit une application $\ell : U \to \mathbb{R}$ associe à chaque arcs de G un poids (ou longueur).

Le problème de l'arbre couvrant de poids minimum de G consiste en la recherche d'un graphe partiel qui soit un arbre et pour lequel la somme des poids des arcs et minimum.

Autrement dit :

Soit $G=(X,U)$ un graphe connexe et $\Gamma = (X,T)$ un arbre couvrant de G.

$$\ell(\Gamma) = \sum_{u\in T} \ell(u) : \text{ est la longueur de l'arbre } \Gamma .$$

Soit E l'ensemble des arbres couvrants du graphe G.

On dit que $\sigma = (X,T)$ est un arbre couvrant à poids minimum de G si :

$$\ell(\sigma) = \min_{\Gamma \in E} \ell(\Gamma)$$

3.2. Algorithmes de recherche d'un arbre couvrant de poids minimum

Soit $G=(X,U)$ un graphe connexe d'ordre n.

On développe 2 algorithmes qui vont construire dans l'ensemble T les arcs d'un arbre partiel de poids minimum de G.

Algorithme de Kruskal

L'algorithme de *Kruskal* permet de construire les arcs d'un arbre couvrant de poids minimum de G en effectuant les choix des arcs de sorte que les poids soient minimum et les cycles soient interdits (fig.3.7):

```
T := Ø
i := 1
Tant que (i < n) Faire
   - Choisir un arc  e_i de poids minimum dans U-T ne
     déterminant aucun cycle avec des arcs de T.
   - T := T ∪ {e_i}
   - i := i+1
Fin Tant que
```

Figure 3.7

Remarque

- L'arbre couvrant de poids minimum d'un graphe G n'est pas unique.
- D'un point de vue informatique, cet algorithme requiert :
  - Qu'on trie initialement les arcs par ordre croissant de leurs poids.
  - Qu'on mette au point une fonction booléenne permettant de dire si un arc e détermine un cycle avec un sous ensemble d'arcs de U.

Exemple 3.5

Appliquons l'algorithme de *Kruskal* au graphe de la figure 3.8 :

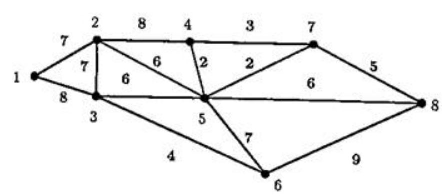
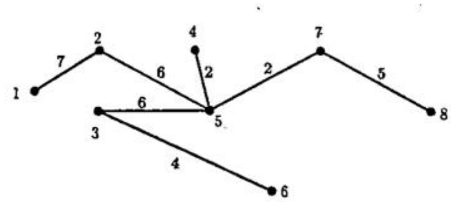


Figure 3.8

Premièrement on trie les arcs dans l'ordre croissant de leurs poids : (5,7), (4,5), (4,7), (3,6), (7,8), (3,5), (5,8), (2,5), (1,2), (2,3), (5,6) ,(2,4), (1,3), (6,8).

Par la suite on choisit dans l'ordre 7 arcs ne constituant pas de cycles avec les arcs précédents.
On aura un arbre couvrant T de poids minimum $L=32$:



Algorithme de Prim

Définition

Soit $v = (x,y)$ un arc d'un graphe $G=(X,U)$.

Le graphe $C_v(G)$ résultant de la "contraction" de l'arc v est obtenu à partir de G en:

- Remplaçant les sommets x et y de G par un sommet unique noté xy.
- L'extrémité initiale (respectivement terminale) d'un arc de $C_v(G)$ est xy ssi l'extrémité initiale (respectivement terminale) de l'arc correspondant dans G est x ou y.

Exemple 3.6

La figure 3.9 représente un exemple de contraction de l'arc $v = (x,y)$:

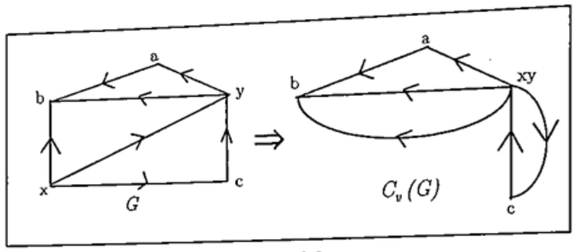


Figure 3.9

Théorème

Soit $G=(X,U)$ un graphe et $v = (x,y) \in U$.

Une condition nécessaire et suffisante pour que G soit un arbre est que $C_v(G)$ soit un arbre.

Exemple 3.7

La figure 3.10 représente un exemple de contraction de l'arc $v = (x,d)$ d'un arbre. La figure 3.11 représente un exemple d'une opération inverse de la contraction de l'arc $v = (x,d)$ d'un arbre.

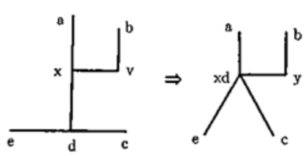


Figure 3.10

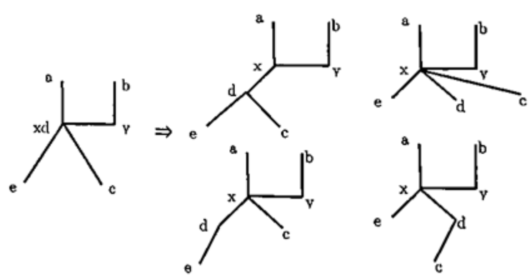


Figure 3.11

Théorème de Prim

Soient $G=(X,U)$ un graphe connexe et $\ell : U \to \mathbb{R}$ une application de pondération de G.

x est un sommet quelconque de G.

S'il existe un arc u_x de G tel que:

$$l(u_x) = \min_{\begin{cases} u \in U \\ x \in \{I(u), T(u)\} \\ u \neq boucle \end{cases}} (l(u))$$

Alors $\exists\, T \subset U$ tel que (X,T) soit un arbre de poids minimum et que $u_x \in T$.

L'algorithme de *Prim* permet de construire les arcs d'un arbre couvrant de poids minimum de G en se basant sur la notion de contraction des arcs et le résultat du dernier théorème (fig.3.12).

T $:=\varnothing$
Tant que *(G comporte plus qu'un sommet)* **Faire**
- *Choisir un sommet x de G.*
- *Déterminer un arc v incident à x tel que :*

$$l(v) = \min_{\begin{cases} u \in U \\ x \in \{I(u), T(u)\} \\ u \neq boucle \end{cases}} (l(u))$$

- $T := T \cup \{v\}$
- $G := C_v(G)$

Fin Tant que

Figure 3.12

Exemple 3.8

On applique l'algorithme de *Prim* au graphe de la figure 3.13 :

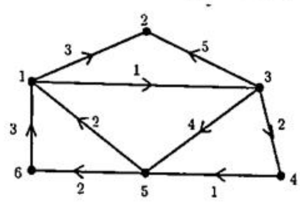
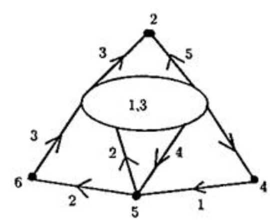


Figure 3.13

Itération 1:

On choisit un sommet quelconque par exemple le sommet 1, l'arc correspondant est *v=(1,3)*. On ajoute l'arc *(1,3)* à *T*.

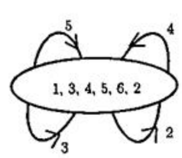
Le graphe $C_v(G)$ est :



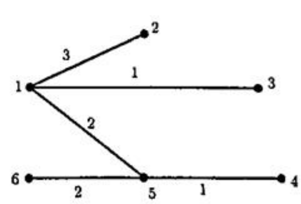
...

Itération 5:

A cette itération le graphe courant contient un seul sommet.



Les arcs ajoutés successivement à *T* constituent les arcs d'un arbre couvrant de poids minimum *L=9* :



4. Arbres binaires et Arborescences

4.1. Arbres binaires

Définition (Arbre binaire)

Un arbre binaire T de n sommets($n \geq 3$) est un arbre comportant un seul sommet de degré 2 et dont tous les autres sommets sont de degré 1 ou 3.

Remarque

- Le sommet de degré 2 dans un arbre binaire est appelé racine de T.
- Un sommet de degré 3 dans un arbre binaire est appelé sommet interne.
- Dans un arbre binaire on dit que la racine a 2 fils, chaque sommet interne a un père et 2 fils et chaque feuille a un père.
- La distance entre deux sommets dans un arbre binaire est la longueur de l'unique chaîne entre ces 2 sommets (chaque arête a une longueur unitaire).

Définition (Niveau d'un sommet dans un arbre binaire)

Dans un arbre binaire T, un sommet x est dit au niveau i $(i \geq 0)$ si x est à une distance i de la racine de T.

Remarque

- La racine d'un arbre binaire est au niveau 0.
- La profondeur L d'un arbre binaire est le niveau maximum associé à ses sommets.

Exemple 3.9

La figure 3.14 représente un arbre binaire de profondeur 4.

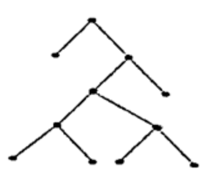


Figure 3.14

4.2. Arborescences

Définition (Racine d'un graphe)

Un nœud a d'un graphe $G = (X,U)$ est une racine s'il existe dans G un chemin joignant a à tout autre sommet du graphe.

Définition (Arborescence)

Un graphe $G = (X,U)$ de n sommets $(n \geq 2)$ est une arborescence de racine a si :

- Le sommet a est une racine de G.
- G est un arbre.

Remarque

- Une arborescence est un arbre mais la réciproque est fausse en général.
- Le concept d'arborescence est essentiellement orienté.

Théorème (Caractérisation des arborescences)

Soit $G_{(X,U)}$ un graphe d'ordre n $(n \geq 2)$.

Les conditions suivantes sont équivalentes et caractérisent une arborescence de racine a :

1) G est un arbre admettant a comme racine.
2) $\forall x \in X$, il existe un chemin unique dans G joignant a à x.
3) G admet a comme racine et minimal au sens des arcs pour cette propriété.
4) G est connexe et de plus $\begin{cases} d_G^-(a) = 0 \\ d_G^-(x) = 1 \quad \forall x \in X, x \neq a \end{cases}$

5) G est sans cycle et de plus $\begin{cases} d_G^-(a) = 0 \\ d_G^-(x) = 1 \ \ \forall x \in X, x \neq a \end{cases}$

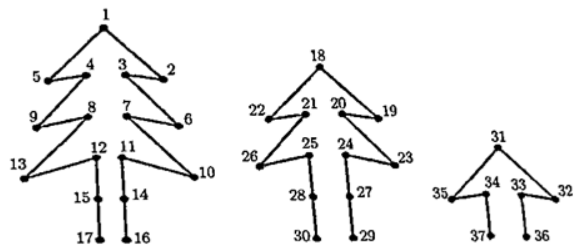
6) G admet a comme racine et il est sans cycle.

7) G admet a comme racine et possède *(n-1)* arcs.

5. Exercices

Exercice 3.1

Etant donné le graphe suivant :



1- Donner la nature de G.

2- Ce graphe contient un arbre partiel ?

Exercice 3.2

Soit $F = (X,U)$ une forêt d'ordre n contenant p composantes connexes($p \geq 2$).

Montrer que le nombre d'arcs de F est $(n-p)$ arcs.

Exercice 3.3

Montrer que le nombre d'arbres construits avec n sommets numérotés est $T_n = n^{n-2}$.

Exercice 3.4

Donner une représentation de tous les arbres qu'on peut construire avec 3,4 sommets numérotés.

Exercice 3.5

Soit $G = (X,U)$ un graphe connexe de n sommets et m arêtes et T un arbre partiel de G.

Donner le nombre de cycles fondamentaux relativement à T.

Exercice 3.6

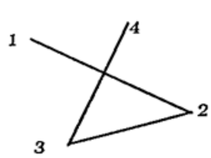
Montrer qu'un arbre sur n sommets (n≥3) ne peut pas être un graphe régulier.

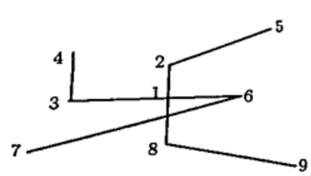
Exercice 3.7

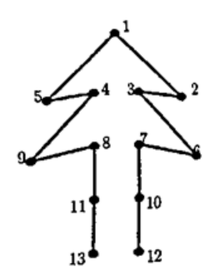
Donner le nombre d'arborescences qu'on peut construire avec n sommets numérotés (n≥2).

Exercice 3.8

Donner le codage de *Prufer* pour les arbres suivants :







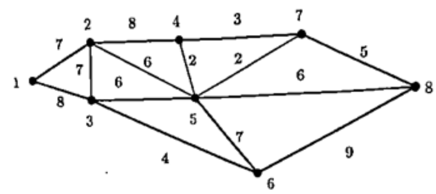
Exercice 3.9

Sachant que la suite S représente le codage de *Prufer*, donner une représentation de l'arbre correspondant dans les cas suivants :

$$S = \{1\}\ ,\ S = \{1,1,2,2,3,3\}\ , S = \underbrace{\{1,1,\ldots,1\}}_{98\ fois}\ , S = \{1,2,\ldots,n\}$$

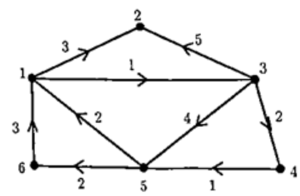
Exercice 3.10

En utilisant l'algorithme de *Kruskal* donner un arbre couvrant de poids minimum du graphe suivant:



Exercice 3.11

En utilisant l'algorithme de *Prim* donner un arbre couvrant de poids minimum du graphe suivant:



Exercice 3.12

Soit E_n l'ensemble des graphes $G_{(X,U)}$ non orientés d'ordre n $(n \geq 3)$ (avec $X=\{1,2,\ldots,n\}$) sans arêtes parallèles qui vérifient la propriété suivante :

$$\forall k \in X - \{1,n\} : \{k-1, k+1\} \notin U$$

1- Calculer pour n=3,4,5 : $\underset{G_{(X,U)} \in E_n}{Max} |U|$

2- Montrer que : $\forall n \geq 3$: $\underset{G_{(X,U)} \in E_n}{Max} |U| = \frac{n(n-1)}{2} + 2$

Soit F_n l'ensemble des graphes connexes $G_{(X,U)}$ non orientés d'ordre n $(n \geq 3)$ (avec $X=\{1,2,\ldots,n\}$) sans arêtes parallèles.

3- Calculer en fonction de n : $\underset{G_{(X,U)} \in F_n}{Min} |U|$

Exercice 3.13

1ère Partie : Arbre Général

Soit $T=(X,U)$ un arbre d'ordre N $(N \geq 3)$

Soit n un entier tel que $2 \leq n < N$.

Supposons que l'arbre T contient un seul sommet de degré n et les autres sommets sont de degrés 1 ou $n+1$.

1- Montrer que le nombre F de feuilles de cet arbre est : $F = \frac{N(n-1)+1}{n}$

2- En déduire le nombre I de sommets de degré $(n+1)$ en fonction de N et n.

3- On pose $N=1000$ et $n=999$. Sachant que le sommet de degré n a le numéro 1 :

3.1- Donner une représentation de cet arbre.

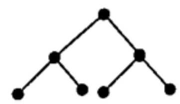
3.2- Donner le codage de *Prüfer* de T.

3.3- Proposer une coloration pour l'arbre T.

$2^{ème}$ Partie : Arbre binaire saturé

On dit qu'un arbre binaire est saturé si seul le dernier niveau (le niveau maximal) contient des feuilles (les niveaux intermédiaires ne contiennent pas des feuilles).

Exemple : un arbre binaire saturé



Soit $T=(X,U)$ un arbre binaire saturé d'ordre N $(N \geq 3)$ et de profondeur h.

4- Tracer T pour $N=15$. Donner sa profondeur.

5- En utilisant la $1^{ère}$ partie donner le nombre F de feuilles et le nombre I de sommets internes de T.

6- Montrer par récurrence sur i que le niveau i de l'arbre T contient 2^i sommets pour tout $i \geq 0$.

7- En déduire que le nombre de feuille F est : $F= 2^h$.

8- Montrer que : $h=log_2(n+1)-1$

Chapitre 4 : Problème du plus court chemin

1. Définitions et propriétés d'un plus court chemin

1.1. Position du problème

Définition (Longueur d'un chemin)

Etant donné un graphe orienté $G=(X,U)$ et une application $d : U \to \mathbb{R}$ qui à chaque arc fait correspondre sa longueur.

La longueur d'un chemin C dans le réseau $R=(X,U,d)$ sera égale à la somme des longueurs des arcs qui constituent le chemin C :

$$l(C) = \sum_{u \in C} d(u)$$

Remarque

Un chemin qui ne comporte aucun arc a une longueur nulle.

Définition (Plus court chemin)

Le problème du plus court chemin entre deux sommets i et j consiste à trouver un chemin $C(i,j)$ de i à j dont la longueur soit minimum.

La longueur de ce plus court chemin quant il existe est appelée la distance de i à j :

$$d(i,j) = \begin{cases} \infty \text{ si il n'existe pas de chemin de } i \text{ à } j \\ min\ \{l(C) \ / \ C \text{ est un chemin de } i \text{ à } j\} \end{cases}$$

Remarque

Il est important de préciser ce qu'on entend par la recherche du plus court chemin dans un réseau entre 2 sommets i et j :

- On peut rechercher le plus court chemin entre i et j.
- On peut rechercher le plus court chemin élémentaire entre i et j.

Généralement ces deux problèmes sont différents.

Exemple 4.1

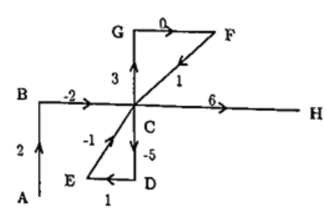


Figure 4.1

- $\nexists$ de chemin de F à B.
- Le plus court chemin élémentaire entre A et H est de longueur 6.
- $\nexists$ de plus court chemin entre A et H (fig.4.1).

Définition (Circuit absorbant)

Un circuit C tel que : $\sum_{u \in C} d(u) < 0$ est appelé circuit absorbant.

Théorème (Condition d'existence d'un PCC)

Pour qu'un chemin de longueur minimum joignant un sommet i à un sommet j dans un réseau $R=(X,U,d)$ existe il faut et il suffit que :

- L'ensemble Y des sommets qui sont à la fois des descendants de i et des ascendants de j soit non vide ($Y=Descendant(i) \cap Ascendant(j)$).
- Le sous réseau construit sur Y ne contienne pas de circuits absorbants.

1.2. Plus courts chemins et distances

Définition

Soit un réseau $R=(X,U,d)$ sans circuits absorbants et admettant le sommet s comme racine.

A chaque sommet x, on associe la longueur $\Pi(x)$ d'un plus court chemin de s à x dans R :

$$\Pi(x) = d(s,x)$$

Théorème

On a : $\Pi(T(u)) - \Pi(I(u)) \leq d(u), \forall u \in U$
$\Pi(s) = 0$

Preuve

Soit $u = (i,j)$ tel que: $\Pi(j) - \Pi(i) > d((i,j))$

Par conséquent : $d(s,j) > d(s,i) + d(u)$

Soient $C_1(s,i)$ un plus court chemin de s à i et $C_2(s,j)$ un plus court chemin de s à j.

Soit C_3 le chemin de s à j constitué du chemin C_1 et de l'arc u.

On vérifie que: $l(C_2) > l(C_3)$

Ce qui signifie que C_2 n'est pas un plus court chemin de s à j. Ce qui est absurde.

Remarque

- Le graphe (X,U') tel que $U' = \{u \in U \mid \Pi(T(u)) - \Pi(I(u)) = d(u)\}$ admet le sommet s comme racine.
- Tout arc d'un plus court chemin de s à n'importe quel sommet appartient à U'.
- Si $\Pi(T(u)) - \Pi(I(u)) < d(u)$, u n'est sur aucun plus court chemin entre s à $T(u)$.
- Tout chemin de s à un sommet quelconque x dans (X,U') est un plus court chemin de s à x.
- Si $\Pi(T(u)) - \Pi(I(u)) = d(u)$ alors il existe un plus court chemin entre s et $T(u)$ contenant u.

2. Algorithmes de recherche des plus courts chemins

2.1. Introduction

Les algorithmes de résolution du problème des plus courts chemins seront différents suivant :

- Les propriétés du graphe.
- Les pondérations des arcs sont positives ou nulles : $d(u) \geq 0, \forall u \in U$.
- G est sans circuits.
- G et $d(u)$ sont quelconques.

Et suivant le problème considéré :

- Recherche des plus courts chemins d'un sommet à un autre.
- Recherche des plus courts chemins d'un sommet à tous les autres.
- Recherche des plus courts chemins de n'importe quel sommet à n'importe quel autre.

2.2. Algorithme de Bellman

Hypothèses

- $R=(X,U,d)$ ne comporte pas de circuits.
- s est le seul sommet tel que $\Gamma_s^{-1} = \varnothing$.

But

Recherche des plus courts chemins du sommet s à tous les autres.

Principe

Il consiste à rechercher les distances de proche en proche sur les sommets du réseau. On ne calcule la plus courte distance de s à un sommet x que si on a déjà calculé les plus courtes distance de s à tous les prédécesseurs de x.

L'ensemble S utilisé dans l'algorithme désigne les sommets pour lesquels on a déjà calculé les plus courtes distances (fig.4.2).

Données : X, U, I, T, d, s
Résultats : Π, A
$S := \{s\}$; $\Pi(s) := 0$; $A(s) := \emptyset$;
Tant que $(\exists j \notin S$ *tel que* $\Gamma_j^{-1} \subset S$ *et* $\Gamma_j^{-1} \neq \emptyset)$ *Faire*
$\Pi(j) := \min_{(u/T(u)=j)} (\Pi(I(u)) + d(u))$
$A(j) := \{u\}$;
$S = S \cup \{j\}$
Fin Tant que

Figure 4.2

Remarque
L'arborescence des plus courts chemins n'est pas forcément unique car il existe généralement dans $A(x)$ plusieurs arcs ayant la même extrémité terminale x.

Exemple 4.2
On applique l'algorithme de *Bellman* au graphe de la figure 4.3 pour déterminer les plus courts chemins du sommet s aux autres sommets du graphe :

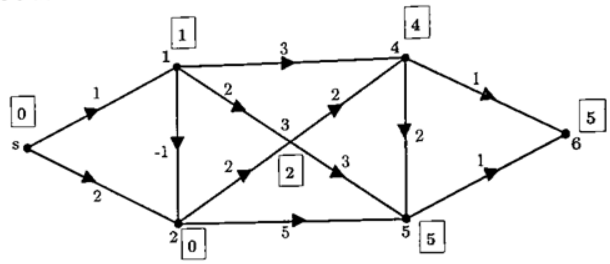


Figure 4.3

On initialise l'ensemble S à s.

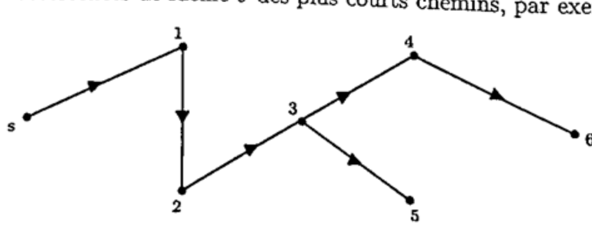
Itération 1:
On choisit $j=1$.
On a : $\Pi(1) = 1$, $A(1) = \{(s,1)\}$, $S = \{s,1\}$

Itération 2:
On choisit $j=2$.
On a : $\Pi(2) = 0$, $A(2) = \{(1,2)\}$, $S = \{s,1,2\}$

...

Itération 6:
On choisit $j=6$.
On a $\Gamma_6^{-1} = \{4,5\} \subset S$
On a : $\Pi(6) = 5$, $A(6) = \{(4,6)\}$, $S = \{s,1,2,3,4,5,6\}$

Les valeurs des plus courts chemins sont indiquées sur le graphe (fig 4.3).
On obtient ainsi 4 arborescences de racine s des plus courts chemins, par exemple:



2.3. Algorithme de Dijkstra

Hypothèses

$R=(X,U,d)$ est tel que les longueurs sont positives ou nulles: $d(U) \subset \mathbb{R}^+$

But

Recherche des plus courts chemins d'un sommet à ses descendants.

Principe

Calcul des plus courtes distances de proche en proche (fig.4.4).

```
Données : X,U,I,T,d,s
Résultats : Π ,A,s_racine
S:={s} ; Π (s):=0 ; A(s):=∅ ; x_pivot:=s ; Π (x):=+ ∞ ,∀x∈ X ( x≠ s) ;
Tant que (S≠X et Π(x_pivot) < +∞) Faire
    Pour tout u∈U tel que : I(u)=x_pivot et T(u) ∉ S
      x:=T(u) ;
      Si ( Π(x) > Π(x_pivot)+d(u)) Alors
        Π(x):=Π(x_pivot)+d(u) ;
        A(x):={u} ;
      Fin Si
    Fin Pour
  Choisir x∉ S tel que Π(x)=min (Π(y))
  x_pivot:=x ;
  S:=S∪ {x_pivot} ;
Fin Tant que
Si ( S=X) Alors s_racine:=Vrai ;
Sinon s_racine:=Faux ;
Fin Si
```

Figure 4.4

Remarque

- L'arborescence des plus courts chemins est unique car il existe dans $A(x)$ un seul arc sortant du sommet pivot et entrant dans le sommet x.
- Si la variable booléenne S_{racine} est à la valeur *Faux* alors il existe dans le graphe un sommet inaccessible à partir du sommet s.

Exemple 4.3

On applique l'algorithme de *Dijkstra* au graphe de la figure 4.5 pour déterminer les plus courts chemins du sommet s aux autres sommets du graphe :

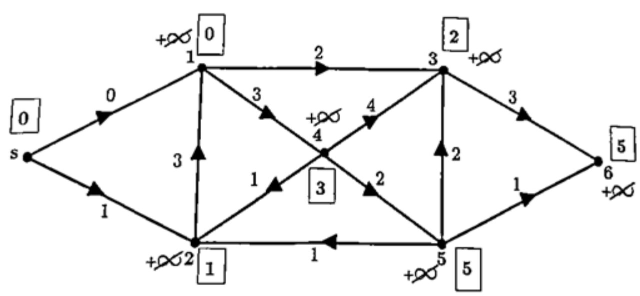


Figure 4.5

On initialise l'ensemble S et le sommet pivot à s, les longueurs des plus courts chemins sont initialisées à $+\infty$ sauf pour le sommet s où $\Pi(s) = 0$.

Itération 1:

Soit l'arc $u=(s,1)$, on a : $I(u)=x_{pivot}$ et $T(u) \notin S$

On a : $\Pi(1) = 0$, $A(1) = \{(s,1)\}$

Pour l'arc $u=(s,2)$

On a : $\Pi(2) = 1$, $A(2) = \{(s,2)\}$

$x_{pivot} = 1$, $S = \{s,1\}$

Itération 2:

Soit l'arc $u=(1,3)$.

On a : $\Pi(3) = 2$, $A(3) = \{(1,3)\}$

Pour l'arc $u=(1,4)$

On a : $\Pi(4) = 3$, $A(4) = \{(1,4)\}$
$x_{pivot} = 2$, $S = \{s,1,2\}$

...

Itération 5:

Soit l'arc $u=(4,5)$.

On a : $\Pi(5) = 5$, $A(5) = \{(4,5)\}$

On a : $x_{pivot} = 6$, $S = \{s,1,2,3,4,6\}$

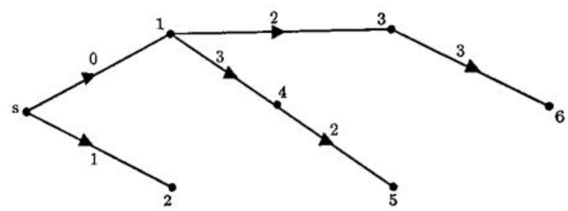
Itération 6:

On a : $x_{pivot} = 5$, $S = \{s,1,2,3,4,6,5\}$

Le sommet s est une racine de ce graphe.

Les valeurs $\Pi(x)$ des plus cours chemins de s à x sont indiquées sur le graphe (fig 4.5).

On obtient ainsi une arborescence unique des plus courts chemins de la racine s aux autres sommets:



3. Exercices

Exercice 4.1

Soit un réseau $R=(X,U,d)$ sans circuits absorbants et admettant le sommet s comme racine.

Montrer que : $\Pi(T(u)) - \Pi(I(u)) \leq d(u), \forall u \in U$

Exercice 4.2

Soit un réseau $R=(X,U,d)$ sans circuits absorbants et admettant le sommet s comme racine.

Montrer que :

- Si $\Pi(T(u)) - \Pi(I(u)) = d(u)$ alors il existe un plus court chemin entre s et $T(u)$ contenant u.
- Si $\Pi(T(u)) - \Pi(I(u)) < d(u)$ alors u n'est sur aucun plus court chemin entre s et $T(u)$.

Exercice 4.3

Soit un réseau $R=(X,U,d)$ sans circuits absorbants et admettant le sommet s comme racine.

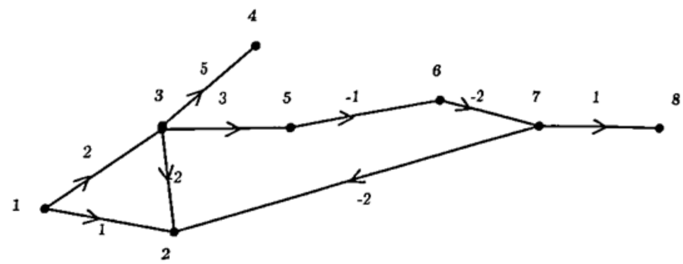
$G' = (X,U')$ est le graphe partiel de G engendré par $U' = \{u \in U \ / \ \Pi(T(u)) - \Pi(I(u)) = d(u)\}$

Montrer que :

- Le graphe $G' = (X,U')$ admet le sommet s comme racine.
- Tout arc d'un plus court chemin de s à n'importe quel sommet appartient à U'.
- Tout chemin de s à un sommet quelconque x dans (X,U') est un plus court chemin de s à x.

Exercice 4.4

Soit $G = (X,U)$ le graphe ci-dessous :



Quelle est l'hypothèse qui permet d'appliquer l'algorithme de *Bellman* ?

En se basant sur cet algorithme :

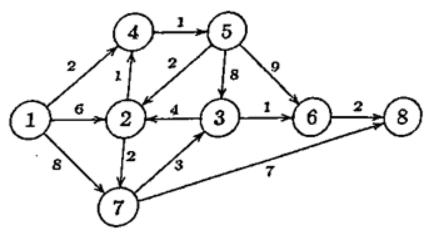
1- Donner la longueur du plus court chemin $\Pi(x)$ de *1* à x pour tout x de X.

2- Donner le nombre d'arborescences obtenues correspondant aux plus courts chemins de 1 à x pour tout x de X.

3- Tracer ces arborescences.

Exercice 4.5

Soit $G = (X,U)$ le graphe ci-dessous :



En utilisant l'algorithme de *Dijkstra*:

Donner la longueur du plus court chemin $\Pi(x)$ de 1 à x pour tout x de X.

Tracer l'arborescence correspondant aux plus courts chemins de 1 à ses descendants.

Chapitre 5 : Problème du flot maximal

1. Définition et position d'un problème de flot

Le problème de flot maximal consiste à transporter la quantité maximale possible d'une origine à une destination donnée sans dépasser les capacités des liaisons.

1.1. Flot dans un réseau de transport

Définition (Réseaux de transport)

On appelle réseau de transport un graphe fini de sommets sans boucles comportant une entrée s et une sortie t.

Remarque

- Depuis le sommet s il existe un chemin vers tout autre sommet x : s est appelé source.
- De tout sommet x il existe un chemin vers le sommet t : t est appelé puits.
- Tout arc u est valué par entier positif $c(u)$ nommé capacité de l'arc u qui représente une capacité de transport associée à la liaison figurée par cet arc.
- Les capacités de transport peuvent être des tonnages disponibles sur des bateaux, des camions, des wagons, ou encore des débits dans des canalisations, voies de transmission, etc.
- Etant donné un réseau de transport le problème consiste à acheminer une quantité maximale de s à t en tenant compte des capacités de transport.

Définition (Flot)

Un flot dans un réseau $G_{(X,U)}$ est un vecteur d'entiers $\Phi = (\varphi_{i,j})$ associés aux arcs tel que :

- $0 \le \varphi(u) \le c(u) : \forall u \in U$ (Si $u = (i,j)$ alors $\varphi(u) = \varphi_u = \varphi_{ij}$ est appelé flux sur l'arc u).
- En tout sommet i différent de la source s et du puits t, on a la loi de conservation suivante : la somme des flux arrivant sur le sommet i est égale à la somme des flux partant du sommet i :

$$\sum_{j\in\Gamma^+(i)} \varphi(i,j) = \sum_{j\in\Gamma^-(i)} \varphi(j,i) : \forall i \in X - \{s,t\}$$

- La valeur du flot noté $V(\Phi)$ est la somme des flux arrivant sur le puits t :

$$V(\Phi) = \sum_{j\in\Gamma^+(s)} \varphi_{sj} = \sum_{j\in\Gamma^-(t)} \varphi_{jt}$$

Remarque

Le problème de flot maximal peut être formulé sous forme d'un programme linéaire :

$$Max\ \mathrm{v} = \sum_{j\in\Gamma^+(s)} \varphi_{sj}$$

$$\begin{gathered} 0 \le \varphi_{ij} \le c_{ij} : \forall (i,j) \in U \\ \sum_{j\in\Gamma^+(i)} \varphi_{ij} = \sum_{j\in\Gamma^-(i)} \varphi_{ji} \ : \forall i \neq s,t \\ \sum_{j\in\Gamma^+(s)} \varphi_{sj} = \sum_{j\in\Gamma^-(t)} \varphi_{jt} \end{gathered}$$

1.2. Chaine améliorante

Définition (chaine améliorante)

Une chaine améliorante est une chaine élémentaire μ allant de la source s au puits t (d'origine s et d'extrémité t) telle que: aucun arc direct ne soit saturé et que les flux des arcs indirects soient strictement positifs.

Remarque

- Si l'on parcourt une chaîne de s à t, les arcs directs sont parcourus dans le sens de leur orientation, tandis que les arcs indirects sont parcourus en sens inverse de leur orientation.
- Un arc saturé est un arc dont la quantité de flux associée est égale à la capacité de cet arc.

2. Algorithme de Ford-Fulkerson

2.1. Principe de l'algorithme

L'algorithme de *Ford-Fulkerson* permet de déterminer une chaine améliorante et améliorer le flot sur cette chaîne. Cet algorithme est basé sur l'alternance de deux phases principales:

Initialisation : $\Phi = 0$

1) Phase de marquage (Recherche d'une chaine améliorante)
2) Phase d'augmentation (Améliorer le flot sur la chaîne trouvée en phase de marquage)

Les deux phases sont répétées jusqu'au moment où il n'existe plus de chaîne augmentant.

2.2. Phase de marquage

La procédure de marquage permet de déterminer une chaîne améliorante (fig.5.1):

Marquer la source s par le signe +
Tant que cela est possible //on s'arrête si le puits t est marqué ou si ce n'est pas possible
Choisir un sommet x non marqué vérifiant l'une des 2 propriétés :
Si : $\exists y \in X$ *tel que : ((y est marqué) et* $(y \in \Gamma^{-}(x))$ *et* $(\varphi(y,x) < c(y,x))$ *Alors*
Marquer + le sommet x
Si : $\exists y \in X$ *tel que : ((y est marqué) et* $(y \in \Gamma^{+}(x))$ *et* $(\varphi(x,y) > 0)$ *Alors*
Marquer - le sommet x
P(x) $\leftarrow$ *y // Le prédécesseur de x dans le marquage est y.*

Figure 5.1

Remarque

- Si le puits t est marqué on s'arrête et le flot actuel n'est pas maximal.
- Si le puits t est marqué on retrouve une chaîne améliorante.

2.3. Phase d'augmentation

L'algorithme suivant correspond à la phase d'augmentation et il permet d'améliorer la valeur du flot sur une chaine améliorante (fig.5.2).

Tant qu'il existe μ *, une chaine améliorante Faire*
Augmenter le flux sur la chaîne améliorante μ *de la manière suivante :*
- $\delta^{+} = \min\{c_u - \varphi_u\}$ *, où u est un arc direct de* μ.
- $\delta- = \min\{\varphi_u\}$ *, où u est un arc indirect de* μ.
- $\delta = \min\{\delta^{+}, \delta^{-}\}$
- *Pour tout arc direct u Faire :* $\varphi_u \leftarrow \varphi_u + \delta$
- *Pour tout arc indirect u Faire:* $\varphi_u \leftarrow \varphi_u - \delta$

Figure 5.2

Remarque

- Dans la phase de marquage si le puits t n'est pas marqué, il n'existe pas de chaîne améliorante.
- Dans la phase de marquage si le puits t est marqué, il existe une chaîne améliorante : une telle chaîne est obtenu, à l'issue de l'application de la procédure de marquage, en calculant récursivement $p(t)$ jusqu'au sommet s ($p(x)$ est le sommet qui a permis de marquer x).
- Lorsqu'il n'existe plus de chaîne améliorante, le flot est optimal.

- On a $\delta^+ > 0$ car aucun arc direct n'est saturé.
- On a $\delta^- > 0$ car aucun arc indirect n'a son flux nul.

2.4. Application

Dans ce paragraphe nous proposons un problème pratique résolu par l'application de l'algorithme de Ford-Fulkerson.

Exemple 5.1

Soient trois châteaux, A, B et C alimentant quatre villages D,E,F et G.

- Le château A bénéfice d'une alimentation et d'une réserve capables de débiter *45 l/s.*
- Le château B peut seulement débiter *25 l/s.*
- Le château C peut débiter *20 l/s.*

Plusieurs canalisations existent et leur débit maximal en l/s est mentionné, pour chacune sur le tableau ci-dessous.

- Le village D aurait besoin d'un débit de *30 l/s.*
- Le village E : *10 l/s.*
- Le village F : *20 l/s.*
- Le village G : *30 l/s.*

On demande d'établir la meilleure alimentation possible.

Canalisation	*(A,D)*	*(A,E)*	*(A,G)*	*(B,D)*	*(B,E)*	*(B,F)*	*(C,F)*	*(C,G)*
Débit maximal	10	15	20	20	5	15	10	10

Solution

On ajoute au graphe représentant les canalisations avec leurs débits, une source O et un puits P, toutes deux fictives, on obtient un réseau de transport (fig.5.3).

Les arcs (O,A), (O,B) et (O,C) ont comme capacité les disponibilités respectives en A, B et C.

Les arcs $(D,P),(E,P),(F,P)$ et (G,P) sont valués par les besoins respectifs en D,E,F et G.

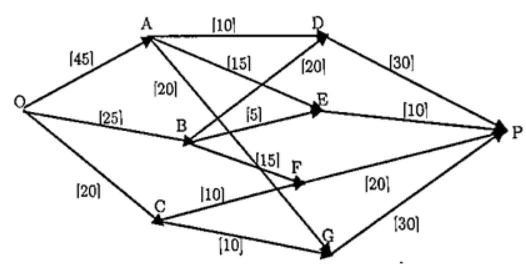


Figure 5.3

Le problème se ramène alors à faire passer un flot de valeur maximale de O vers P (ou sur un arc imaginaire dit «arc de retour» qui reviendrait de P vers O)

Appliquons l'algorithme :

On choisit un flot initial Φ sur ce réseau de transport.

Ce flot vérifie : $\begin{cases} \forall u \in U : 0 \le \varphi_u \le c_u \\ \forall i \in X - \{O,P\} : \sum\limits_{j \in \Gamma^+(i)} \varphi_{ij} = \sum\limits_{j \in \Gamma^-(i)} \varphi_{ji} \end{cases}$

$v(\Phi) = 80(l/s)$

Les arcs saturés sont indiqués en gras (fig.5.4).

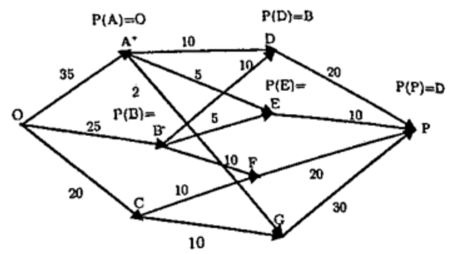
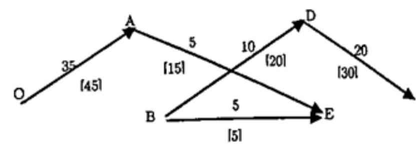


Figure 5.4

Phase de marquage

- O est marqué +
- P est marqué + et $P(A)=O$
- E est marqué + et $P(E)=A$
- B est marqué − et $P(B)=E$
- D est marqué + et $P(D)=B$
- Le puits P est marqué + et $P(P)=D$

On s'arrête et on peut augmenter le flot sur la chaîne améliorante suivante :



Phase d'augmentation

Itération1

$\delta^+ = 10$, $\delta^- = 5$, $\delta = 5$

$\varphi'_{OA} = 40$, $\varphi'_{AE} = 10$, $\varphi'_{BD} = 15$, $\varphi'_{DP} = 25$, $\varphi'_{BE} = 0$

On obtient un flot de valeur $V(\Phi') = 85$

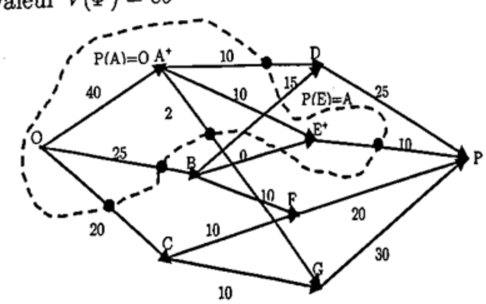


Figure 5.5

Phase de marquage

- La source O est marqué +
- A est marqué + et $P(A)=O$
- E est marqué + et $P(E)=1$

Aucun autre sommet ne peut être marqué et par conséquent le flot obtenu est de valeur maximale.

Remarque

Dans l'exemple précédent, on peut justifier qu'il s'agit d'une solution optimale (sans démonstration de l'algorithme):

Considérons une ligne fermée (ou coupe), à l'intérieur de laquelle sont tous les sommets marqués.

- Vers l'extérieur de cette courbe ne sortent que des arcs saturés.

- Vers l'intérieur de cette courbe ne rentrent que des arcs de flux nuls.

Il est évident que ce n'est pas possible de :

- Faire sortir un flot de valeur supérieur à 85 (tous les arcs sortants sont saturés).
- Réduire le flot entrant (ce flot est nul).

Et puisque la courbe contient la source O, c'est bien le flot maximal qui part de la source.

2.5. Aspect théorique de l'algorithme de Ford-Fulkerson

Le théorème de *Ford-Fulkerson* permet de justifier le dernier algorithme.

Définition (Coupe)

Une coupe dans le réseau de transport $G=(X,U)$ est définie par la donnée de 2 ensembles S et $\delta^+(S)$ tels que :

- S est un sous ensemble non vide de sommets.
- La source s appartient à S.
- $\delta^+(S)$ est l'ensemble d'arcs sortant de S : $\delta^+(S) = \{(i,j) \in U : i \in S, j \in X - S\}$

Définition (Capacité d'une coupe)

La capacité d'une coupe S est la somme des capacités de ses arcs : $C(S) = \sum_{i\in S, j\in \bar{S}} c_{ij}$

Théorème

Pour toute coupe $(S,\bar{S})$ et tout flot Φ, On a : $V(\Phi) = \sum_{i\in S, j\notin S} \varphi_{ij} - \sum_{i\in S, j\notin S} \varphi_{ji}$

Ce qui signifie que la valeur du flot est la somme des flux sortants de S diminuée de la somme des flux entrants dans S.

Preuve

On a : $V(\Phi) = \sum_{i\in\Gamma^+(s)} \varphi_{si}$

$V(\Phi) = \sum_{i\in\Gamma^+(s)} \varphi_{si} - \sum_{i\in\Gamma^-(s)} \varphi_{is}$

Soit $i \in S$ tel que $i \neq s$:

On a : $\sum_{j\in\Gamma^+(i)} \varphi_{ij} - \sum_{j\in\Gamma^-(i)} \varphi_{ji} = 0$

Donc : $V(\Phi) = \sum_{i\in\Gamma^+(s)} \varphi_{si} - \sum_{i\in\Gamma^-(s)} \varphi_{is} + \sum_{i\in S, i\neq s}\left(\sum_{j\in\Gamma^+(i)} \varphi_{ij} - \sum_{j\in\Gamma^-(i)} \varphi_{ji}\right)$

Il s'ensuit que : $V(\Phi) = \sum_{i\in S}\left(\sum_{j\in\Gamma^+(i)} \varphi_{ij} - \sum_{j\in\Gamma^-(i)} \varphi_{ji}\right)$

Après développement des sommes, on a : $V(\Phi) = \underbrace{\sum_{i,j\in S} \varphi_{ij} - \sum_{i,j\in S} \varphi_{ji}}_{=0} + \sum_{i\in S, j\notin S} \varphi_{ij} - \sum_{i\in S, j\notin S} \varphi_{ji}$

Et par Conséquent : $V(\Phi) = \sum_{i\in S, j\notin S} \varphi_{ij} - \sum_{i\in S, j\notin S} \varphi_{ji}$

Propriété (Inégalité : Capacité d'une coupe-valeur d'un flot)

Pour toute coupe S (de capacité $C(S)$) et tout flot Φ (de valeur $V(\Phi)$), On a : $C(S) \geq V(\Phi)$

Propriété caractéristique d'un flot maximal

S'il existe une coupe $\bar{S}$ et un flot $\bar{\Phi}$ tels que :

$C(\bar{S}) = V(\bar{\Phi})$ alors $\bar{\Phi}$ est un flot de valeur maximale et $\bar{S}$ est une coupe de capacité minimale.

Propriété de Ford-Fulkerson

La capacité minimale des coupes est égale à la valeur maximale de flots.

Preuve de l'algorithme

L'algorithme s'arrête lorsqu'il n'y a plus de chaîne améliorante.

Soit Φ le dernier flot obtenu par l'application de l'algorithme.

Dans ce cas il n'existe plus de chaîne améliorante, et par conséquent le puits p n'est pas marqué après l'application de la procédure de marquage.

Soit S l'ensemble des sommets marqués et $\bar{S}$ son complémentaire (les sommets non marqués).

$s \in S$ et $p \in \bar{S}$ sont non vide et définissent bien une coupe.

Pour tout arc *(i,j)* tel que : $(i,j) \in \delta^+(S)$

On a : $i \in S$ et $j \notin S$

L'arc *(i,j)* est forcément saturé, ce qui implique $\varphi_{ij} = c_{ij}$

Pour tout arc *(j,i)* avec $i \in S$ et $j \notin S$, on a nécessairement $\varphi_{ji} = 0$ (car j n'a pas été marqué)

Il s'ensuit que : $C(S) = \sum_{i \in S, j \notin S} c_{ij} = \sum_{i \in S, j \notin S} \varphi_{ij} - \sum_{i \in S, j \notin S} \varphi_{ji}$

Finalement: $C(S) = V(\Phi)$

En utilisant la propriété caractéristique le flot Φ est maximal et la coupe S est de capacité minimale.

3. Flot de valeur maximale à coût minimal

3.1. Définition et position du problème

Dans de nombreux problèmes, outre les capacités limitées de transport, le coût d'acheminement d'une certaine quantité de produit doit être pris en compte.

Le problème de la recherche d'un flot maximal à coût minimal permet de prendre en compte ce second objectif dans le cadre d'un réseau de transport.

Pour un flot optimal (de valeur V^* maximale), il existe fréquemment de nombreux flots différents de même valeur V^*, parmi ceux-ci, il est intéressant d'en trouver un de moindre coût.

Ce problème se formalise de la manière suivante :

- R est un réseau de transport.
- s et p désignent respectivement la source et le puits.
- A chaque arc *(i,j)* sont associées deux valeurs positives $[c_{ij}, \ p_{ij}]$:
  - c_{ij} est la capacité associée à l'arc *(i,j)*.
  - p_{ij} est le coût unitaire associée à l'arc *(i,j)*.
- Le coût du flux φ_{ij} circulant le long de l'arc *(i,j)* est : $\varphi_{ij} \cdot p_{ij}$

Définition (Coût d'un flot)

Le coût d'un flot Φ est la somme des coûts sur tous les arcs du réseau : $\sum_{(i,j)} \varphi_{ij} \cdot p_{ij}$

3.2. Graphe d'écart associé à un flot

Définition (Graphe d'écart)

Soit R un réseau de transport.

Soit $\Phi = (\varphi_{ij})$ un flot sur R.

Nous supposons que le graphe R est antisymétrique.

On définit le graphe d'écart G_Φ^e associé au flot Φ de la manière suivante :

- Le graphe d'écart G_Φ^e et le réseau de transport R ont les mêmes sommets.

- Pour tout arc *(i,j)* de *R*, les arcs du graphe d'écart et leurs valuations sont obtenus de la façon suivante :

Si $0 < \varphi_{ij} < c_{ij}$ Alors : $\begin{cases} (i,j) \text{ est un arc de } G^e_\Phi \\ \text{La valuation} : r_{ij} = c_{ij} - \varphi_{ij} \end{cases}$ Et $\begin{cases} (j,i) \text{ est un arc de } G^e_\Phi \\ \text{La valuation} : r_{ji} = \varphi_{ij} \end{cases}$

Si $\varphi_{ij} = 0$ Alors : $\begin{cases} (i,j) \text{ est un arc de } G^e_\Phi \\ \text{La valuation} : r_{ij} = c_{ij} \end{cases}$

Si $\varphi_{ij} = c_{ij}$ Alors : $\begin{cases} (j,i) \text{ est un arc de } G^e_\Phi \\ \text{La valuation} : r_{ji} = \varphi_{ij} \end{cases}$

Remarque

- Si le graphe *R* n'est pas antisymétrique (les deux arcs *(i,j)* et *(j,i)* existent), on peut remplacer l'arc *(i,j)* par deux arcs *(i,k)* et *(k,j)*, tous deux de capacité c_{ij} et de coût $\frac{1}{2}\, p_{ij}$.
- Pour le flot nul : $\Phi = (0,\dots,0)$, le graphe d'écart et le réseau de transport coïncident.

Propriété (chaîne améliorante-chemin de s à p)

A une chaîne améliorante dans *R* correspond un chemin de la source au puits dans G^e_Φ et réciproquement.

Propriété (flot maximal-chemin de s à p)

Un flot est maximal si et seulement s'il n'existe pas de chemin de *s* à *p* dans G^e_Φ.

Définition (valeurs des coûts)

Soit l'arc *(i,j)* du réseau de transport.

Dans le graphe d'écart G^e_Φ :

- Si l'arc *(i,j)* existe $(\varphi_{ij} < c_{ij})$ alors le coût de *(i,j)* et $+p_{ij}$
- Si l'arc *(j,i)* existe $(\varphi_{ij} > 0)$ alors le coût de *(j,i)* et $-p_{ij}$

3.3. Algorithme de Roy

Théorème de Roy

Un flot Φ est de coût minimal parmi les flots de valeur $V(\Phi)$, si et seulement si, il n'existe pas de circuit de coût strictement négatif dans G^e_Φ.

Remarque

Le flot Φ n'est pas nécessairement de valeur maximale.

Algorithme de Roy

Cet algorithme permet de calculer un flot maximal de coût minimal (fig.5.6).

$\Phi := (0,0,\dots,0)$

$G^e_\Phi := R$

Tant que (Il existe un chemin de s à p dans G^e_Φ *) Faire*

- *Déterminer un chemin C de s à p de coût minimal*
- $\delta = \min_{(i,j)\in C} r_{ij}$
- *Pour tout arc (i,j) de C :*
  - *Si (i,j) est un arc de R alors* $\varphi_{ij} \leftarrow \varphi_{ij} + \delta$
  - *Si (j,i) est un arc de R alors* $\varphi_{ji} \leftarrow \varphi_{ji} - \delta$
- *Tracer le graphe d'écart* G^e_Φ *du flot ainsi modifié.*

Fin Tant que

Figure 5.6

Remarque

- A chaque étape de l'algorithme un flot Φ' est calculé à partir d'un flot Φ en utilisant une chaîne améliorante.
- Le choix de la chaîne améliorante utilisée fait que le graphe d'écart G_Φ^e associé n'a pas de circuit de coût strictement négatif.
- La détermination d'un chemin de coût minimal peut se faire en utilisant l'algorithme de *Dijkstra*.

Exemple 5.2

Soit le réseau de transport suivant (fig.5.7):

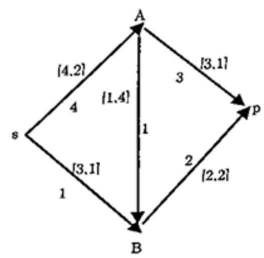


Figure 5.7

Le flot initial est de valeur 5 et de coût 20.

Le graphe d'écart associé à ce flot (fig.5.8):

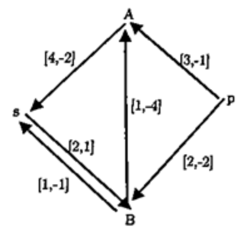


Figure 5.8

Le circuit *((A,S),(S,B),(B,A))* est de coût -5.
Donc ce flot n'est pas de coût minimal.

Appliquons l'algorithme de *Roy* (Les flèches épaisses correspondent à chaque étape au chemin de coût minimal de *s* vers *p*).

Soit le réseau de transport (fig.5.9):

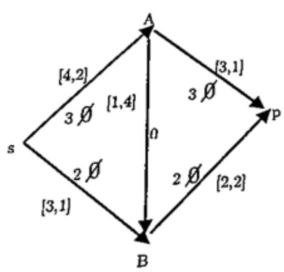


Figure 5.9

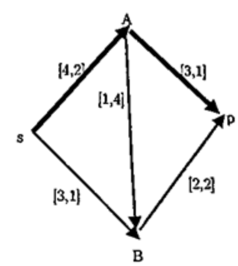


Figure 5.10

Flot initial: $\Phi^{(0)} = 0$, $G^e_{\Phi^{(0)}} = R$ (fig.5.10)

Itération 1 :

$C^{(1)} = ((s,A),(A,p))$ de coût 3 et $\delta^{(1)} = 3$.

Le flot $\Phi^{(1)}$ est de valeur 3 et de coût 9.

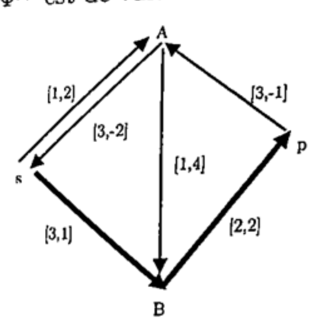


Figure 5.11

Itération 2 :

$C^{(2)} = ((s,B),(B,p))$ de coût 3 et $\delta^{(2)} = 2$.

Le flot $\Phi^{(2)}$ est de valeur maximale 5 et de coût minimal 15.

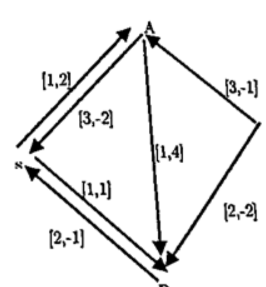


Figure 5.12

Il n'existe plus de chemin de *s* à *p*.

4. Exercices

Exercice 5.1

Montrer que dans un réseau de transport, la capacité minimale des coupes est égale à la valeur maximale de flots.

Exercice 5.2

Montrer que s'il existe une coupe $\tilde{S}$ et un flot $\tilde{\Phi}$ tels que :

$C(\tilde{S}) = V(\tilde{\Phi})$ alors $\tilde{\Phi}$ est un flot de valeur maximale et $\tilde{S}$ est une coupe de capacité minimale.

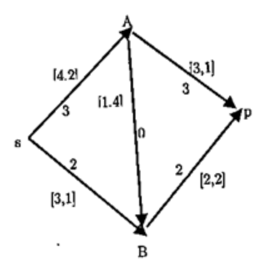
Exercice 5.3

Soient $(S,\bar{S})$ une coupe et Φ un flot.

Montrer que la valeur du flot Φ est la somme des flux sortants de S diminuée de la somme des flux entrants dans S.

Exercice 5.4

Soit le réseau de transport suivant:



Appliquer l'algorithme de *Roy* pour déterminer un flot maximal de cout minimal.

Exercice 5.5

Trois dépôts A, B, C disposent respectivement de 30, 20 et 45 tonnes de marchandise ; cinq destinations D, E, F, G et H en demandent des quantités respectives de 10, 25, 20, 25 et 15 tonnes. Des camions, faisant route entre les points désignés, offrent les disponibilités (tonnages) ci-contre.

	D	E	F	G	H
A	5	5		20	10
B		20	10		5
C	10	5	10	10	10

Etablir le meilleur plan de transport

Exercice 5.6

Un graphe de $n=5$ sommets et $m=8$ arcs est décrit par la liste des successeurs :

I	1	2	3	4	5
d_i^+	2	2	3	0	1

Où d_i^+ le demi-degré extérieur du sommet i, est le nombre d'arcs partant du sommet i.

I	1	2	3	4	5	6	7	8
ext(j)	2	5	1	5	1	2	5	4

Le tableau *ext(j)* liste, dans l'ordre lexicographique, les extrémités terminales des arcs issus du sommet 1 puis celles du sommet 2, etc. Ainsi le sommet 1 a deux successeurs : ce sont les 2 premiers éléments du tableau *ext(j)*, donc les sommets 2 et 5.

1- Tracer ce graphe. Montrer, en détail, qu'il s'agit d'un réseau de transport (la capacité de chaque arc étant donnée à la question suivante).

2- On associe à chaque arc u_j une capacité et un flux :

j	1	2	3	4	5	6	7	8
Capa(j)	1	3	2	5	5	3	4	12
flux(j)	1	3	2	2	2	3	4	9

Vérifier que les flux proposés forment bien un flot sur ce réseau de transport.
Déterminer, obligatoirement à l'aide de l'algorithme approprié, si ce flot est maximal. Sinon, l'optimiser. Donner la valeur du flot maximal; indiquer une coupe minimale, et rappeler sa signification concrète.