REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LARECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

Support de Cours et de Travaux Dirigés

Théorie des Graphes

Module Informatique 3

2ème année cycle préparatoire

Par: Dr. Nour El-Houda BENALIA

Table de matière

Chapitre Introductif

Motivation, Introduction et Historique

1. Motivations

Pour résoudre de nombreux problèmes concrets, on est amené à tracer sur le papier des petits dessins qui représentent (partiellement) le problème à résoudre. Bien souvent, ces petits dessins se composent de points et de lignes continues reliant deux à deux certains de ces points. On appellera ces petits dessins des graphes, les points des sommets et les lignes des arcs ou arêtes, selon que la relation binaire sous-jacente soit orientée ou non.

2. Introduction et Historique

La théorie des graphes est un très vaste domaine (sous domaine de la recherche opérationnelle), en évolution constante tant du point de vue des applications que de celui des recherches fondamentales et constitue aujourd'hui un corpus de connaissances très important. Ce cours ne constituera donc qu'une introduction à cette théorie.

La théorie des graphes est née au dix-huitième siècle en essayant de trouver des solutions à des exercies considérés comme étant des curiosités mathématiques.

En 1736 Euler présenta une communication dans laquelle il proposait une solution au célèbre problème des ponts de Kônigsberg. Le problème posé était le suivant. Deux deltas C et D sur la rivière Pregel à Kônigsberg (Actuellement Kaliningrad) étaient reliées entre elles ainsi qu'aux rivages A et B à l'aide de sept ponts comme le montre la figure ci-dessous.

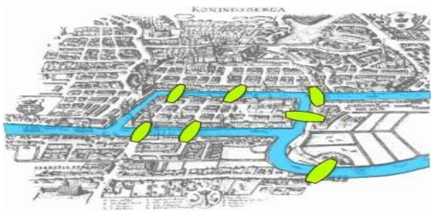


Figure: Les ponts de Kônigsberg.

Le problème consistait, à partir d'un point quelconque A, B, C, ou D, à traverser chacun des ponts une fois et une seule et à revenir au point de départ. Euler représenta cette situation à l'aide d'un graphe et démontra que ce problème n'admet pas de solution.

Deux autres problèmes d'importance pour la théorie des graphes furent également proposés.
Le premier est la conjecture des quatre couleurs qui affirme que quatre couleurs suff-isent pour colorier n'importe quelle carte plane telle que les pays ayant une frontière commune soient de couleurs différentes. Ce problème est resté très longtemps sans solution. Il a fallu attendre jusqu'en 1976 pour que Appel et Haken prouvent ce théorème en réduisant le

problème à un nombre fini de situations particulières et en trouvant une solution pour chacune d'entre elles à l'aide d'un ordinateur.

Le second problème est dû à Sir Hamilton. En 1859, il inventa un casse-tête qui consiste en un dodécaèdre régulier en bois (un polyèdre à 12 faces et 20 sommets), chaque face étant un pentagone régulier. Trois arêtes sont donc issues de chaque sommet. Un clou est fiché sur chaque sommet marqué du nom de vingt grandes villes mondiales. Le casse-tête consiste à enrouler une ficelle passant une fois et une seule fois par chacune des villes (sommets). Bien que la solution de ce problème soit aisée à obtenir, personne n'a encore trouvé de condition nécessaire et suffisante de l'existence d'un tel chemin (appelé chemin Hamiltonien) dans un graphe quelconque.

La théorie des graphes s'est alors développée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales. Depuis le début du XXe siècle, elle constitue une branche à part entière des mathématiques.

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

Les graphes constituent donc à la fois une théorie et un outils qui permettent de modéliser une grande variété de problèmes en se ramenant à l'étude de graphes. Les derniers travaux en théorie des graphes sont aussi effectués par des informaticiens, du fait de l'importance que revêt l'aspect algorithmique.

3. Domaines d'application de la théorie des graphes

Les graphes (et par conséquent la théorie des graphes) sont utilisés dans de nombreux domaines. On peut donner quelques exemples:

- Les réseaux de communication : réseaux de routes représentés par une carte routière, réseaux de chemin de fer, de téléphone, de relais de télévision, réseaux électriques, réseaux des informations dans une organisation, etc... ;
- La gestion de la production : graphes potentiels-étapes plus connu sous le nom de graphes PERT ["Programme Evaluation and Research Task" ou "Programme Evaluation Review Technique"] ;
- L'étude des circuits électriques : Kirchhof, qui a étudié les réseaux électriques, peut être considéré comme un des précurseurs de cette théorie ;
- La chimie, la sociologie et l'économie : la notion de clique est un exemple de l'implication de la théorie des graphes dans ces disciplines.

4. Plan du cours

Dans ce cours, nous verrons différents outils de la théorie des graphes sans apporter de justifications **mathématiques très détaillées et rigoureuses**.

Dans ce cours, l'accent est mis sur les aspects fondamentaux de la théorie des graphes, les principaux problèmes ainsi que les solutions relatives propres à chaque problématique. Dans ce qui ce suit, on présente

- Introduction
- Notions de bases, représentation des graphes, cheminements, connexité, parcours Hamiltonien et Eulérien;
- Arbres;
- Recherche du plus court chemin;
- Recherche du flot maximum;
- Arbres couvrant de poids minimum;
- Programmation linéaire.

5. Exemples

- **Chemin le plus court / le plus long**

Soit un ensemble de villes et des chemins directs reliant ces villes entre elles. Le problème dit "du plus court chemin" consiste à trouver pour une ville de départ donnée et une ville d'arrivée donnée le chemin le plus court qui relie ces deux villes. Le problème peut également être de trouver un chemin le plus court pour chaque couple de villes. Pour certains problèmes, trouver le plus long chemin entre deux points peut être intéressant.

- **Flot maximum**

Soit des châteaux d'eau ayant un débit constant. Ils desservent un certain nombre de villes, chacune ayant des besoins quantifiés constants. L'eau est acheminée à travers des conduits dont le débit maximum est connu. Le problème est de trouver un moyen de satisfaire au mieux les demandes de chaque ville. En d'autres termes, essayer d'apporter le plus d'eau vers les villes.

- **Flot de coût minimum**

Il s'agit d'un problème semblable à celui du flot maximum mais on suppose en plus qu'un coût fonction du débit est associé à l'utilisation d'un conduit. Le problème devient alors de satisfaire les villes mais de la manière la moins onéreuse.

- **Sac à dos**

Un randonneur prépare son sac à dos pour partir en excursion. Bien entendu, il veut éviter d'avoir un sac trop lourd et décide de se limiter dans le choix des objets qu'il emporte afin de ne pas dépasser un certain poids. Cependant, il veut emporter le maximum de choses utiles.
Pour cela, il affecte une valeur quantitative à chaque objet en plus de son poids (plus la valeur est importante, plus le randonneur juge l'objet important). Le problème peut donc se formuler de la manière suivante: trouver l'ensemble des objets dont la somme des utilités est maximum tout en ne dépassant pas un poids fixé.

- **Voyageur de commerce**

Un voyageur de commerce doit démarcher dans un certain nombre de villes. Il connaît bien entendu la distance qui sépare les villes entre elles. Cependant, le voyageur de commerce veut perdre le moins de temps possible dans ses déplacements. Le problème est donc de trouver un chemin qui passe par toutes les villes une et une seule fois et qui soit le court possible.

- **Affectation**

Des modifications de postes sont effectuées dans une entreprise. Plusieurs personnes doivent être affectées à de nouveaux postes. Ainsi, chacun classe par ordre de préférence les postes qu'il veut occuper. Le problème ici est d'attribuer à chaque personne un poste tout en essayant de satisfaire au mieux le souhait de chacun.

Dans tous ces exemples, il existe une méthode simple pour résoudre le problème. En effet, il suffit d'énumérer toutes les possibilités et d'en dégager la ou les meilleures. Cependant, on s'aperçoit que plus le problème est compliqué en terme d'éléments mis en jeu, plus le nombre de possibilités croît de manière non pas linéaire (proportionnelle) mais plutôt exponentielle. Par exemple, le problème d'affectation présenté précédemment avec 100 personnes a 100! (100 x 99 x 98 x ... x 1) solutions. Le simple fait de rajouter une personne dans le problème va multiplier par 101 le nombre de solutions.

Généralement en recherche opérationnelle (en théorie des graphes automatiquement), on a souvent à traiter des problèmes dont le nombre de solutions devient rapidement difficile à imaginer. Bien que les exemples vus ici soient petits, il faut bien comprendre qu'en réalité, on sera confronté à des problèmes de taille beaucoup plus importante. Ce qui explique que l'on cherche des méthodes toujours plus efficaces pour résoudre les problèmes.

Partie 01:
Chapitres

Chapitre 1
Concepts de base de la théorie des graphes

Objectifs:

- Savoir comment représenter graphiquement un problème;
- Comprendre la notion de connexité dans un graphe;
- Comprendre la notion de chemin dans un graphe.

Dans ce cours: Vous allez voir:

1) Les définitions;
2) La structure d'un graphe;
3) Les graphes particuliers;
4) Les cheminements dans un graphe;
5) La connexité;
6) La forte connexité;
7) La mise en ordre d'un graphe connexe ou la recherche d'un circuit;
8) Les cheminements euleriens;
9) Les chemins hamiltoniens;
10) L'algorithme de K-coloration d'un graphe.

1. Introduction

Les graphes représentent de manière simple et naturelle des relations entre les objets. Ils constituent un moyen très pratique pour représenter différents types d'objets et différentes situations de la vie courante. Ils peuvent aussi représenter la structure d'un ensemble complexe avec ses éléments et les relations entre eux. Les outils mathématiques et les algorithmes mis au point en théorie des graphes permettent de résoudre une multitude de problèmes, tels que les problèmes de cheminement, d'ordonnancement, d'affectation, etc.

Ce chapitre présente les aspects fondamentaux de la théorie des graphes.

2. Définition d'un graphe et ses différentes représentations

Un graphe est une structure comportant un ensemble non vide X d'éléments appelés sommets et un ensemble A d'éléments appelés arcs (ou arêtes).

2.1 Définition d'un graphe

Un graphe G =(X, A) est le couple constitué par:

1. un ensemble X de sommets X = $\{x_1, x_2, \ldots\ldots x_n\}$ et
2. un ensemble d'arêtes A = $\{a_1, a_2, \ldots\ldots\ldots\ldots a_p\}$ qui sont les éléments du produit cartésien
   $X*X = \{(x,y)/ x\in X, y \in X\}$

Exemple
Soit X= { 1, 2, 3, 4, 5, 6, 7, 8 }) l'ensemble des sommets, et A= { (1,2), (1,3),(2,3), (2,4), (2,5), (2,6), (4,6), (6,7),(3,7) } L'ensemble des arêtes.

Card (X) = 8 et Card (A) = 9

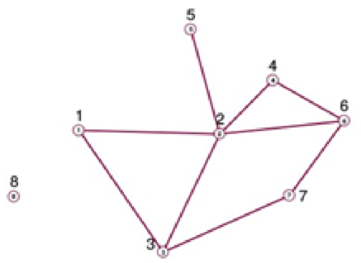


Figure 1.1 Représentation d'un graphe.

2.2 Orientation d'un graphe

Certains graphes possèdent une propriété utile pour la modélisation de systèmes particuliers, il s'agit de l'orientation. Le trait qui relie deux sommets peut être orienté (on parle alors d'arc, A= E, $G = (X,E)$) ou non (on parle alors d'arête, A=U,$G = (X,U)$). En représentation graphique, l'orientation est représentée par une flèche.

2.2.1 Graphe orienté

Un graphe orienté G = (X, U) est défini par:

– $X = \{x_1, x_2, \ldots, x_n\}$, l’ensemble fini de sommets, $n > 1$ et
– $U = \{u_1, u_2, \ldots, u_m\}$, l’ensemble fini d’arcs.
Chaque élément $u_i \in U$ est une paire ordonnée de sommets, $u_i = (x, y)$. x est appelé extrémité initiale de u_i et y est l’extrémité terminale. U peut être vide.
Exemple: La Figure 1.2 (a)

2.2.2 Graphe non orienté

Un graphe non orienté G = (X, E) est défini par :
– $X = \{x_1, x_2, \ldots, x_n\}$, l’ensemble fini de sommets, $n > 1$.
– $E = \{e_1, e_2, \ldots, e_m\}$, l’ensemble fini d’arêtes.
Chaque élément $e_i \in E$ est une paire non ordonnée de sommets, $e_i = (x, y)$. x et y sont appelés extrémités de e_i .
E peut être vide.
Les sommets représentent les objets et les arcs ou les arêtes représentent les liens entre les différents objets.
Exemple: La Figure 1.2 (b)

2.2.3 Représentation graphique

On représente généralement un sommet par un point ou par un cercle. Un arc est représenté par une flèche et une arête par un trait qui peuvent être courbés.

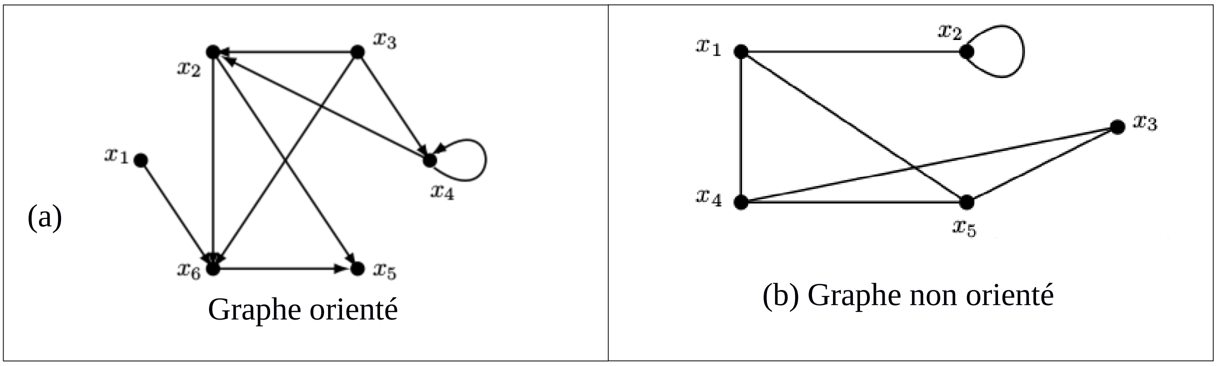


Figure 1.2 Représentation de deux graphes, orienté et non orienté.

2.3 Terminologie

La terminologie suivante s’applique à tout graphe G :

- Les deux sommets u et v sont appelés extrémités de l’arête (u,v). S’il s’agit d’un arc, u et v désignent respectivement l’extrémité initiale et terminale de l’arc. Le sommet u est le prédécesseur de v et v est le successeur de u.
- Les arcs ayant même extrémités sont appelés arcs parallèles.
- Un arc de la forme (v,v) est une boucle.
- Un graphe est simple s’il n’a ni arcs parallèles ni boucles.
- Un graphe sans arcs est un graphe vide ($G = (X, A) / A = \emptyset$).
- Un graphe sans sommets est un graphe nul ($G = (X, A) / X = \emptyset$).
- Un graphe ayant un seul sommet est un graphe trivial ($G = (X, A) / Card(X) = 1$).
- Des arcs sont adjacents s’ils ont un sommet en commun à l’une des extrémités.
- Deux sommets u et v sont voisins ou adjacents s’ils sont connectés par un arc.
Le voisinage d’un sommet désigne l’ensemble de ses sommets voisins.

Soit $x \in X$, un sommet du graphe G, on appelle voisin de x tout sommet $y \in X$ différent de x et qui est adjacent à x. Ainsi, on définit le voisinage V(x) comme suit :
– $V(x) = \{y \in X - \{x\} / \{x, y\} \in E\}$ pour les graphes non orientes.
– $V(x) = V^+(x) \cup V^-(x)$ où $V^+(x) = \{y \in X - \{x\} / (x, y) \in U\}$ et
$V^-(x) = \{y \in X - \{x\} / (y, x) \in U\}$. (graphe orienté)
– $V^+(x)$ (resp. $V^-(x)$) est appelé ensemble des voisins externes (resp. internes) de x.
- Un graphe G est dit complet si et seulement si $\forall$ x, y $\in$ X, $(x,y) \notin A \Rightarrow (y,x) \in A$. c'est-à-dire que tous les couples de sommets qu'il est possible de choisir (chacun étant pris une seule fois) sont présents dans A.

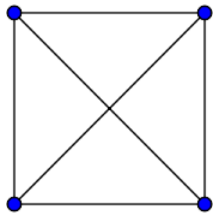


Figure 1.3 Un exemple de graphe complet.

2.4 Notion d'ordre et de degrés

Les deux nombres d'un graphe sont l'ordre et le degré.

2.4.1 Ordre d'un graphe

Dans la définition ci-dessus d'un graphe, le nombre n représente l'ordre du graphe G. il s'agit du nombre de ses sommets. Dans l'exemple de la Figure 1.1, l'ordre du graphe est 4.

2.4.2 Notion de degré

Définition 1 (degré) Soit G = (X,E) un graphe non orienté (resp. G = (X, U) un graphe orienté). A tout sommet x $\in$ X, on peut associer une valeur entière positive ou nulle, notée dG(x), qu'on appelle degré du sommet x. Cette valeur est définie comme suit :
dG(x) = nombre de fois où x est extrémité d'un arc (resp. d'une arête).

2.4.2.1 Dans le cas d'un graphe non orienté

Définition 2 Soit G = (X,E) un graphe non orienté.
Le degré d'un sommet x, noté dG(x), est le nombre d'arêtes ayant x comme extrémité ou le nombre d'arêtes incidentes à ce sommet. Par convention, la boucle est comptée deux fois.

$d^+G(x) = \text{card}\{(x,y) / (x,y) \in E\} + 2 * \text{card}\{(x,x) / x,x) \in E\}$

Exemple Dans l'exemple de la Figure 1.2 (b) :
- $d(x_1) = 3$
- $d(x_3) = 2$

Formule cas non orienté :
Pour tout graphe non orienté G = (X,E), on a : $\sum_{x \in X} dG(x) = 2|E|$

Preuve Chaque arête a exactement deux extrémité ⇒ Elle est comptée deux fois dans le calcul de la somme totald'arêtes.
⇒ La somme totale des degrés est égale a deux fois le nombre total d' arêtes.

2.4.2.2 Dans le cas d'un graphe orienté,

Le degré d'un sommet est calculé comme étant la somme des demi-degrés extérieurs et intérieurs d'un sommet.

Définition 3 Soit G = (X,U) un graphe orienté.
On appelle demi-degré extérieur d'un sommet x ∈ X, la valeur suivante :

$$d^+G(x) = |\{u \in U / I(u) = x\}|$$

- demi-degré extérieur : $d^+(i)$ est le nombre d'arcs dont i est l'extrémité initiale.

On appelle demi-degré intérieur d'un sommet x ∈ X, la valeur suivante :

$$d^-G(x) = |\{u \in U / T(u) = x\}|$$

- demi-degré intérieur : $d^-(i)$ est le nombre d'arcs dont i est l'extrémité terminale.

Le degré d'un sommet x : $dG(x) = d^+G(x) + d^-G(x)$

Exemple Dans l'exemple de la Figure 1.2 (a) : $d^+G(x_2) = 2$ et $d^-G(x_2) = 2$.

Remarque Pour tout graphe, nous avons : $dG(x) > |V(x)|$. Si G est un graphe non orienté simple alors on a : $dG(x) = |V(x)|$, où$V(x)$ est l'ensemble des voisins de x.

Formule cas orienté :

Pour tout graphe oriente G = (X,U), on a :

$$\sum_{x\in X} dG(x) = 2|U| \qquad \text{et} \qquad \sum_{x\in X} d^+G(x) = \sum_{x\in X} d^-G(x) = |U|$$

Preuve Chaque arc a exactement une extrémité initiale et une extrémité terminale ⇒ Chaque arc est comptabilise une fois dans d+ pour son extrémité initiale et une autre fois dans d− pour son extrémité finale. Le nombre total d'arcs ayant une extrémité initiale (resp. terminale) est exactement la somme des demi-degrés extérieurs (resp. intérieurs).

Conséquence : De là, on peut déduire que le nombre de sommets de degrés impairs dans un graphe est toujours pair.

2.4.2.3 Théorème 1 : (Théorème d'Euler)

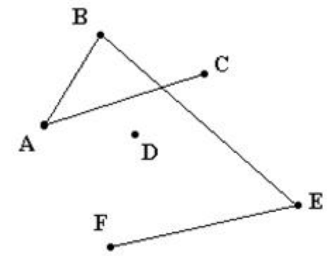
La somme des degrés des sommets est égale au double du nombre de ses arcs.

Remarques :
- un sommet pendant est un sommet dont le degré est 1.
  Si dG(x) = 1 Alors x est dit sommet pendant. (**Dans le graphe ci-dessouss, les sommets F et C sont pendants**).
- Un sommet isolé est un sommet dont le degré est nul .
  Si dG(x) = 0 Alors x est dit sommet isolé. (**Dans le graphe ci-dessouss, le sommet D est un sommet isolé**).
– Un arc (resp. une arête) incident(e) a un sommet pendant est aussi dit(e) arc (arête) pendant(e).(**L' arête E-F est une arête pendante**).
– On appelle degrés minimal d'un graphe G qu'on note par δ(G), le plus petit degré dans le graphe G. $\delta(G) = \min\{dG(x)\}$

$\delta(G) = \min\{dG(x)\} = (dG(D)) = 0$

−On appelle degré maximal d'un graphe G qu'on note par $\Delta(G)$, le plus grand degré dans le graphe G. $\Delta(G) = \max\{dG(x)\}$

$\Delta(G) = \max\{dG(x)\} = dG(A) = dG(B)= dG(E) = 2$



3. Représentations machine d'un graphe

Un certain nombre de représentations existent pour décrire un graphe. On distingue principalement la représentation par matrice d'adjacence, par matrice d'incidence sommets-arcs et par les listes d'adjacence.

3.1 Matrice d'adjacence

A tout graphe simple d'ordre n on associe une matrice $M_A(xi,xj)$ de n lignes et n colonnes dont les éléments sont notés Mij .

$$\boxed{M_{ij}} = \begin{cases} 1 \; si \; (x_i x_j) \in A \\ 0 \; sinon \end{cases}$$

Pour un graphe non orienté $G = (X,E)$, $Mij=Mji$ représente le nombre d'arêtes ayant les sommets i et j comme extrémités.

Remarque 1

- La matrice d'adjacence M_A d'un graphe non orienté est toujours symétrique.
- La somme d'une ligne k = la somme d'une colonne $k = dG(k)$.

Pour un graphe orienté $G=(X, U)$, Mij représente le nombre d'arcs ayant i comme extrémité initiale et j comme extrémité terminale.

Remarque 2:

- Dans le cas d'un graphe valué (dont les arcs ont des valeurs), le « 1 » est remplacé par la valeur de l'arc.

– La somme d'une ligne $i = d^+G(i)$.

- La somme d'une colonne $j = d^-G(j)$.

Exemple : Si l'on considère le graphe orienté de la Figure 1.2, sa représentation par la matrice d'adjacence est comme suit :

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

Remarques 3

– Les coefficients de M_A pour un graphe simple est binaire avec des 0 sur la diagonale.
– Les éléments de M_A pour un multi graphe (graphe contenant des arcs (resp. arêtes) parallèles) sont le nombre d'arcs (arêtes) de chaque sommets.

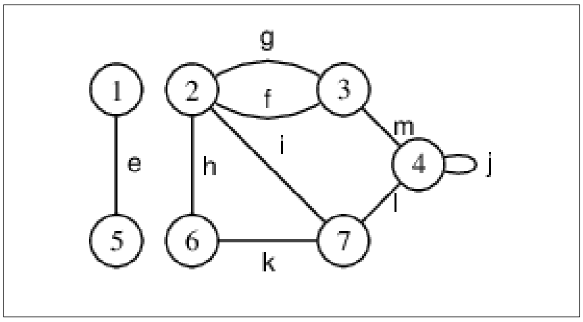
3.2 Matrice d'incidence sommets-arcs

A tout **graphe non oriente** $G = (X,E)$, on peut associer une matrice M de n lignes et m colonnes, *où* n est le nombre de sommets dans G et m est le nombre d'arêtes dans G.
Mij représente le nombre de fois où le sommet *i est incident à l'arête j*. Les éléments de M sont dans $\{0, 1, 2\}$.

$$M_I(i, j) = \begin{cases} 0 \text{ si } a_j \text{ non incident à } i. \\ +1 \text{ si } a_j \text{ incident à i.} \\ +2 \text{ si } a_j \text{ est une boucle} \end{cases}$$

Exemple: La représentation par matrice d'incidencedu graphe non orienté à droite.

	e	f	g	h	i	j	k	l	m
1	1	0	0	0	0	0	0	0	0
2	0	1	1	1	1	0	0	0	0
3	0	1	1	0	0	0	0	0	1
4	0	0	0	0	0	2	0	1	1
5	1	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	1	0	0
7	0	0	0	0	1	0	1	1	0



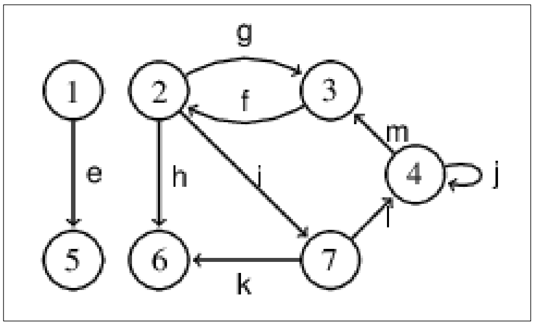
A tout **graphe orienté** $G = (X,U)$, on peut associer une matrice M de n lignes et m colonnes, *où* n est le nombre de sommets dans G et m est le nombre d'arcs dans G.

Chaque élément $M_I(i,j)$ de la matrice a pour valeur :

$$M_I(i, j) = \begin{cases} 0 \text{ si } a_j \text{ non incident à i} \\ +1 \text{ si } a_j \text{ sortant de i} \\ -1 \text{si } a_j \text{ rentrant dans i} \\ +2 \text{ si } a_j \text{ est une boucle} \end{cases}$$

Exemple: La représentation par matrice d'incidencedu graphe orienté à droite.

	e	f	g	h	i	j	k	l	m
1	-1	0	0	0	0	0	0	0	0
2	0	-1	1	1	-1	0	0	0	0
3	0	1	-1	0	0	1	0	0	-1
4	0	0	0	0	0	2	0	-1	1
5	-1	0	0	0	0	0	0	0	0
6	0	0	0	-1	0	0	-1	0	0
7	0	0	0	0	0	-1	1	1	0



Remarques :
- Si deux colonnes $j1$ et $j2$ sont identiques alors les arêtes $j1$ et $j2$ sont parallèles.
- Si un élément $Mij = 2$ alors l'arête j est une boucle.
- Pour toute colonne correspondant à un arc (excepté pour les boucles), la somme est nulle.

3.3 Liste d’adjacence

Pour un graphe G=(X,A), la liste d’adjacence associée à ce graphe permet de représenter pour chaque sommet, l’ensemble de ses successeurs.
Tous les arcs émanant d’un même sommet, sont liés entre eux dans une liste. A chaque arc sont donc associés l’extrémité terminale et le pointeur au prochain sommet de la liste.

On crée un tableau LP, de dimension n, qui est destiné à contenir les têtes de liste. Dans ce tableau, pour tout sommet i, on trouve un pointeur vers la liste de ses successeurs. La figure 1.4 illustre un graphe et la liste d’adjacence qui lui est associée.

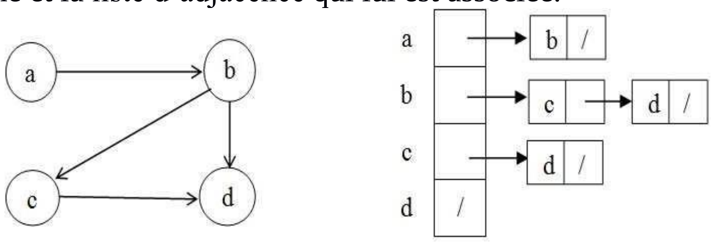


Figure 1.4 Un graphe avec sa liste d'adjacence.

La liste d'adjacence d'un graphe non orienté, est la liste des voisins de chaque sommet.

4 . Graphes particuliers

Il arrive parfois que seul un sous-ensemble de sommets ou d’arcs soit pertinent à la résolution d’un problème. Pour cette raison, nous donnons les définitions de quelques graphes particuliers.

Soit $G = (X,U)$ un graphe orienté (resp. $G = (X,E)$ un graphe non oriente). Soit $A \in X$ un sous ensemble de sommets et $V \in U$ (resp. $V \in E$).

4.1 Sous graphe

Le sous-graphe de G = (X,A) induit par l’ensemble Xs ⊆ X est un graphe *G1 = (Xs, As)* consistant en un sous ensemble de l’ensemble des sommets de X et en tous les arcs reliant les sommets de ce sous ensemble.

Un sous-graphe de G engendré par l’ensemble de sommets A est le graphe :
$Gs = (Xs,Us)$ *où* $Us = \{u \in U/I(u) \in A \text{ et } T(u) \in A\}$ dans le cas oriente.
$Gs = (Xs,Es)$ *où* $Es = \{e = \{x, y\} \in E/x \in A \text{ et } y \in A\}$ dans le cas non oriente

La Figure 1.5 présente un graphe G et deux de ses sous-graphes. Le premier est induit par l’ensemble Xs= {a, b}, le second est induit par Xs’= {b, c, d}.

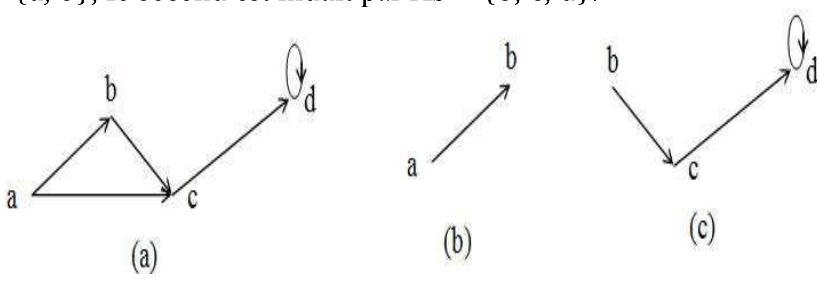


Figure 1.5 Un graphe et quelques uns de ses sous-graphes.

Si le sous-graphe d’un graphe est complet, on dit qu’il forme une ***clique***. Dans la Figure 1.5, lesous-graphe induit par Xs_s est une clique car il est complet.

4.2 Graphe partiel

Le graphe partiel de G = (X,A) induit par l'ensemble Ap ⊆ A est le graphe Gp = (X, A_p) comportant tous les sommets de G et un sous ensemble d'arcs (arêtes) A_p de A.

La partie (a) de la Figure 1.6 illustre un graphe partiel Gp du graphe G de la Figure1.5. Gp est induit par l'ensemble Ap= {(a,b), (b,c), (d,d)}.

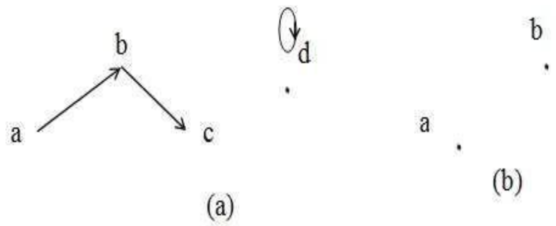


Figure 1.6 Un graphe partiel et un sous-graphe partiel.

4.3 Sous-graphe partiel

Le sous-graphe partiel de G = (X,A) est le graphe Gsp = (Xs , Ap) induit par un sous ensemble Xs de sommets de X et par un sous ensemble d'arcs (arêtes)Ap de A. on peut le considérer comme le sous-graphe d'un graphe partiel.
La partie (b) de la Figure 1.6 illustre un sous-graphe partiel Gsp du graphe G de la Figure 1.5. Gsp est induit par l'ensemble Xs= {a, b} et l'ensemble Ap= {(b,c), (d,d)}.
La Figure 1.7 illustre un graphe G, un graphe partiel de G ainsi qu'un sous graphe.

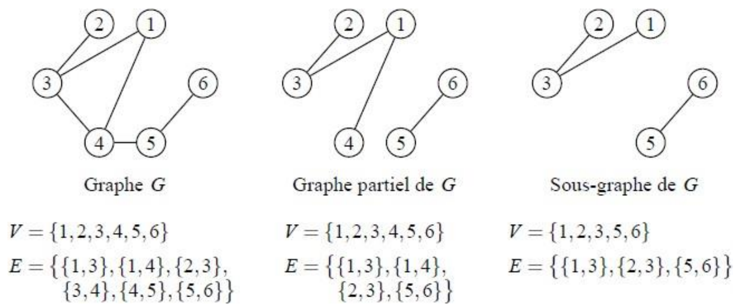


Figure 1.7 illustre un graphe G, un graphe partiel de G ainsi qu'un sous graphe.

4.4 Complément d'un graphe

Le graphe complémentaire de G est note $G = (X, \acute{U})$ (*resp.* $= (X, \acute{E})$) où :
$\acute{U} = \{(x, y) \in X^2/(x, y) \notin U\}$(*resp.* $\acute{E} = \{\{x, y\} \in X^2/\{x, y\} \notin E\}$

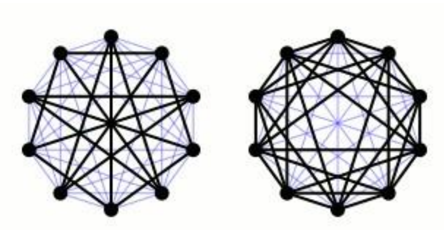


Figure 1.8 Un graphe avec son graphe complémentaire.

4.5 Graphe Simple

Un graphe est dit simple s'il ne contient ni boucles ni arcs parallèles. Si G est simple, on a $dG(x) = |V(x)|$.

4.6 Graphe Complet

Dans le cas orienté :

G est complet ssi $\forall x, y \in X, (x, y) \notin U \Rightarrow (y, x) \in U$

Dans le cas non orienté :

G est complet ssi $\forall x \neq y \in X, \{x, y\} \in E$.
Un graphe simple complet d'ordre n est noté *Kn*.

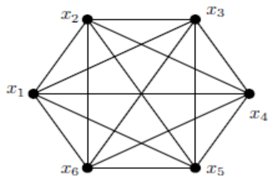


Figure 1.9 Un graphe complet.

4.7 Graphe Régulier

Un graphe G est dit k-regulier
si $\forall x$ sommet de G, on a $dG(x) = k$.
En d'autres termes, $\delta(G) = \Delta(G) = k$.

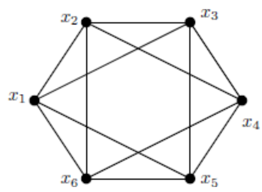


Figure 1.10 Un graphe 4-régulier.

4.8 Graphe Symétrique

Cette notion est spécifique aux graphes orientés. G est symétrique ssi $\forall x, y \in X, (x, y) \in U \Rightarrow (y, x) \in U$

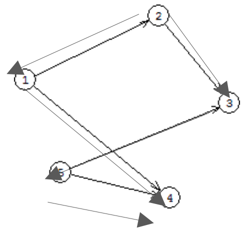
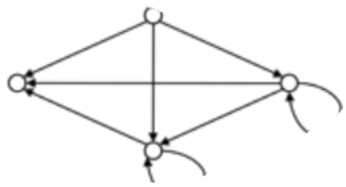


Figure 1.11 Un graphe symétrique.

4.9 Graphe Antisymétrique

Cette notion est spécifique aux graphes orientés.
G est antisymétrique
ssi $\forall x, y \in X$, $(x, y) \in U \Rightarrow (y, x) \notin U$



4.10 Graphe Transitif

Cette notion est spécifique aux graphes orientés. G est transitif ssi $\forall x, y, z \in X, (x, y) \in U$ et $(y, z) \in U \Rightarrow (x, z) \in U$

Figure 1.12 Un graphe transitif.

4.11 Graphe Biparti (bipartite)

G est dit biparti ssi l'ensemble de ses sommets X admet une partition en deux sous ensembles $X1$ et $X2$
avec $X_1 \cap X_2 = \varnothing$ et $X_1 \cup X_2 = X$.

Dans le cas oriente : $\forall (x, y) \in U \Rightarrow x \in X_1$ et $y \in X_2$

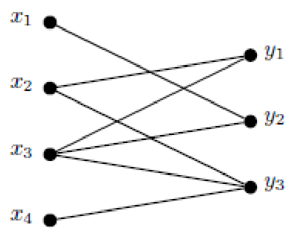


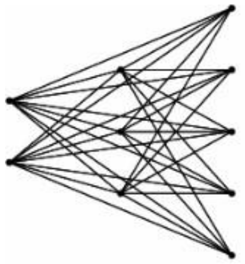
Figure 1.13 – Un graphe biparti.

Dans le cas non oriente : $\forall x, y \in E$ ($x \in X_1$ et $y \in X_2$) *ou* ($x \in X_2$ *et* $y \in X_1$).
G est dit biparti complet ssi G est dit biparti et $\forall x \in X_1$ et $\forall y \in X_2 \Rightarrow (x, y) \in U$.

Un graphe biparti complet et simple $G = (X_1 \cup X_2, U)$ (resp. $G = (X_1 \cup X_2, E)$) avec $|X_1| = p$ et $|X_2| = q$ est note Kp,q.

Théorème
Un graphe est biparti si et seulement s'il ne contient aucun cycle de longueur impaire.

4.12 Graphe Multiparti



G est dit multiparti ssi l'ensemble de ses sommets X admet une partition en p sous ensembles $X_1, X_2, \ldots X_p$ ($p \geq 3$).
Avec $X_i \cap X_j = \varphi$ ($i \neq j$) et $X_1 \cup X_2 \ldots \cup X_p = X$.

Dans le cas oriente : $\forall (x, y) \in U \Rightarrow x \in X_k$ et $y \in X_{k+1}$ ($avec\ 1 \leq k \leq p-1$).

Figure 1.14 – Un graphe Multiparti.

Dans le cas non oriente : $\forall x, y \in E(x \in X_k$ et $y \in X_{k+1})$ ou $(y \in X_k$ et $x \in X_{k+1})$ $avec\ 1 \leq k \leq p-1$

4.13 Graphe Planaire

Un graphe planaire est un graphe qui a la particularité de pouvoir se représenter dans un plan sans qu'aucune arête, courbe ou rectiligne, n'en croise une autre.

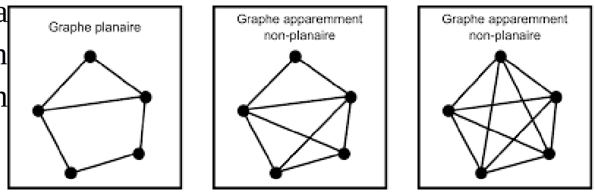


Figure 1.15 – Graphe planaire et non planaire.

Un graphe est dit planaire si on peut le dessiner sur un plan de telle façon que les arêtes ne se coupent pas, en dehors de leurs extrémités.

Ce type de graphe est particulièrement utilisé dans le problèmes de circuits imprimés (ces circuits, construits sur des surfaces planes, constutuent actuellement l'une des limitations des développements de l'informatique).

Remarques:

- Une face d'une graphe planaire est par définition une région du plan limité par des arêtes de telle sorte que deux points arbitraire, dans cette région, reliés par une arete ne rencontrent ni sommet, ni arete.
- La frontière d'une face est l'ensemble des arêtes qui l'entourent.
- Une face infinie est une face illimitée, elle n'admet pas de contour et elle est unique. Les autres faces sont finies
- Deux faces sont dites adjacentes si leurs frontières ont une arete commune.

Important:

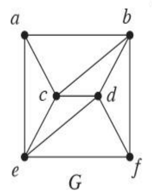
Les graphes planaires vérifient la formule $|X| + F = |E| + 2$, tel que:
F est le nombre de faces (ou régions), $|X|$ le nombre de sommets, et $|E|$ le nombre d'arêtes.

4.14 Stable/Clique

On appelle stable dans un graphe G un sous-ensemble de sommets S ⊆ X tel que le sous graphe engendré par S est formé de sommets isolés (ne contient aucun arc ou arête).

Autrement dit, soit G(X,A) un graphe non orienté. Un sous-ensemble S de G est stable si Gs = (Xs, Us) ne comprend que des sommets non adjacents deux à deux.

Exemple Déterminer une partition du graphe G en sous-ensembles stables S_i



Solution: On peut proposer la partition suivante :
$S_1 = \{a ; d\}$, $S_2 = \{b ; e\}$ et $S_3 = \{c ; f\}$.

La partition n'est pas unique, on peut également proposer :
$S_1 = \{a ; f\}$, $S_2 = \{b ; e\}$, $S_3 = \{c\}$ et $S_4 = \{d\}$.

5. Cheminements dans un graphe

Plusieurs problématiques peuvent être considérées lorsque l'on s'intéresse au cheminement entre deux points d'un graphe. Cette section a pour objectif de définir les concepts de cheminement dans un graphe.

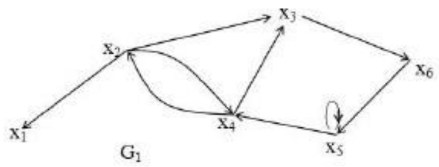


Figure 1.16 Graphe G1.

5.1 Chaines et cycles

Les concepts de chaine et cycle peuvent aussi bien s'appliquer à un graphe orienté qu'à un graphe non orienté.

5.1.1 Chaîne

On appelle chaine dans un graphe non orienté (resp. orienté) $G = (X,E)$ (resp. $G = (X,U)$), une séquence finie de la forme (x0, x1, …..xm). Le sommet x_0 est le sommet initial et xm est le sommet final. Cette séquence est une suite alternée de sommets et d'arêtes (resp. d'arcs), $\mu = x_0x_1 \ . \ . \ . \ x_{k-1}x_k$ (resp. $\mu = x_0x_1 \ . \ . \ . \ x_{k-1}x_k$) Tel que pour tout $1 \leq i \leq k$, x_i et x_{i+1} sont extrémités de l'arête i (resp. de l'arc $i + 1$).
Dans cette séquence, tout arc (arêtes) (x_{i-1}, x_i) a une extrémité en commun avec l'arc (arête) suivant (x_i, x_{i+1}). On dit que μ est une chaine joignant les sommets x_0 et x_k de longueur k.

5.1.2 Chaine élémentaire

Une chaine élémentaire est une chaine qui passe par un sommet au maximum une fois, (si tous les sommets les composants sont distincts).

5.1.3 Chaîne simple

On dit qu'une chaine est simple si tous les arcs ou les arêtes de la composants sont distincts.

Sur la Figure 1.16, la séquence (x_6, x_3, x_2, x_1) représente une chaine élémentaire, alors que la séquence (x3, x4, x2, x4,) n'est pas une chaine élémentaire.

5.1.4 Cycle

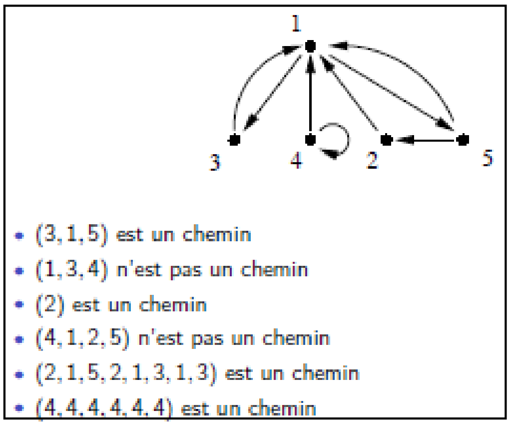
On appelle cycle dans un graphe non orienté (resp. orienté) $G = (X,E)$ (resp. $G = (X,U)$), toute chaine fermée simple : $\mu = x_0x_1 \dots x_{k-1}x_k$ (resp. $\mu = x_0x_1 \dots x_{k-1}x_k$) Tel que k_0, et $x_0 = x_k$. (l'extrémité initiale est à la fois extrémité finale). La séquence de sommets pour un cycle est donc $(x_0, x_1, \dots x_m, x_0)$. On dit que μ est un cycle longueur k.
Sur la Figure 1.16, la séquence (x_3, x_2, x_4, x_3) représente un cycle.

Remarque :
- Notons qu'un même arc ne peut figurer deux fois dans la séquence.

Proposition *Soit $G = (X,E)$ un graphe*
Si le degré minimum $\delta(G) \geq 2$ alors G contient un cycle.
Si G est simple et $\delta(G) \geq k \geq 2$ alors G comporte un cycle de longueur k.

5.2 Chemins et circuits

Les concepts de chemin et circuit sont relatifs aux graphes orientés puisque le sens d'orientation des arcs est essentiel pour ces définitions.



5.2.1 Chemin

Dans un graphe G = (X,A), un chemin allant de x_0 à x_m est une séquence finie de la forme $(x_0, x_1, \dots x_m)$ tel que tout couple (x_{i-1}, x_i) forme un arc orienté dans la même direction, la direction de parcours. Le sommet x_0 est le sommet initial et x_m est le sommet final. Dans cette séquence, pour tout arc (resp. arête) $a_i = (x_{i-1}, x_i)$, l'extrémité terminale x_i coïncide avec l'extrémité initiale de l'arc a_{i+1}. "m" est la longueur du chemin.
Sur la Figure 1.16, la séquence (x_2, x_4, x_3, x_6) représente un chemin allant de x_2 à x_6.

5.2.2 Chemin élémentaire

Un chemin élémentaire est un chemin qui passe par un sommet au maximum une fois.

5.2.3 Chemin simple

On dit qu'un chemin ou une chaine est simple si tous les arcs les composants sont distincts.

5.2.4 Circuit

Un circuit est défini comme un chemin fermé simple, tel que l'extrémité initiale est à la fois extrémité finale. La séquence de sommets pour un circuit est donc (x0, x1, …..xm, x0).
Sur la Figure 1.16, la séquence (x3, x6, x5, x4, x3) représente un circuit.

Remarques
- On dit qu'un cycle ou circuit est élémentaire si tous les sommets qui le composent sont distincts.
- La longueur d'un cycle ou d'un circuit est aussi égale au nombre de sommets formant ce cycle ou circuit.
- Dans un graphe simple, un cycle ou un circuit peuvent être déterminés juste en énumérant la suite des sommets qui les composent.
- Une boucle est un cycle (circuit) élémentaire de longueur 1.

- Toute chaine (ou chemin) élémentaire est aussi simple. L'inverse n'est pas toujours vrai.

6. Connexité

La connexité est une propriété essentielle des graphes surtout pour les problèmes relatifs aux réseaux. Dans ce qui suit, sont étudiées les notions de connexité simple et forte.

Un graphe est **connexe** s'il est possible, à partir de n'importe quel sommet, de rejoindre tous les autres en suivant les arêtes. Un graphe non connexe se décompose en **composantes connexes**. Il existe une chaine joignant chaque paire de sommets *x* et *y* ($x \neq y$).

Il est possible de distinguer deux types de connexité :

- la connexité qui concerne à la fois les graphes orientés et non orientés,
- la forte connexité qui ne concerne que les graphes orientés.

6.1 Connexité simple

Définition 1: Un graphe est dit connexe si et seulement si :

$$\forall\, i, j \in X \begin{cases} i = j \text{ ou} \\ \text{il existe une chaîne reliant } i \text{ et } j \end{cases}$$

En d'autres termes, il doit être possible entre tout couple de sommets de trouver une chaine reliant ces deux sommets.
Si un graphe n'est pas connexe, alors il admet au moins deux composantes connexes.
Le graphe G1 de la Figure 1.17 est un graphe connexe alors que le graphe G2 n'est pasconnexe car il admet deux composantes connexes.

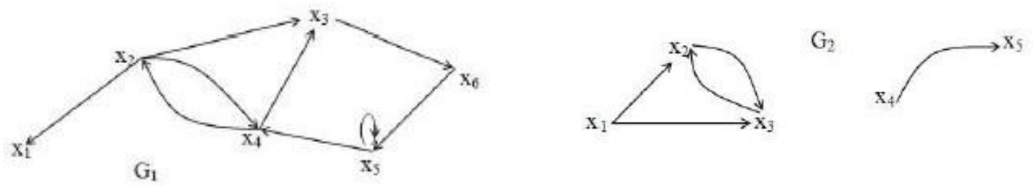


Figure 1.17 G1 est un graphe connexe, G2 est un graphe comportant deux composantes connexes.

6.2 Composante connexe

La **composante connexe** d'un sommet s, notée *CC*(s), est le sous-ensemble de sommets tels qu'il existe une chaîne entre deux sommets quelconques de *CC*(s).
Si un sommet *s*' appartient à la composante connexe d'un sommet *s*, alors la composante connexe de *s*' est égale à celle de *s*.

Définition 2
Soit un graphe $G = (X,E)$ (resp. $G = (X,U)$) :
Une composante connexe CC(s) est un sous-graphe induit maximal connexe. Maximal signifie qu'il n'y a pas de sous-graphe induit connexe plus grand contenant les sommets de CC(s).

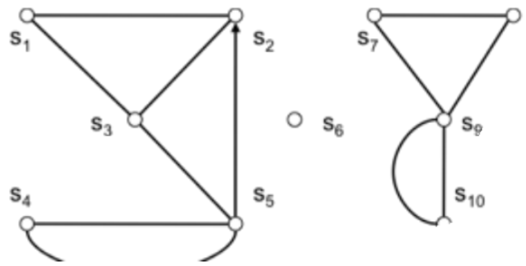
Ou encore, Si le sous graphe engendré par un ensemble de sommets $S \subseteq X$ (G_S) est connexe et le sous graphe engendré par $S \cup x$ et $x \notin S$ n'est pas connexe, alors G_S est une composante connexe de G.
Le sous graphe engendré par un sommet isolé est considéré comme une composante connexe de G.

Exemples
Cet exemple présente les composantes connexes d'un graphe non orienté:

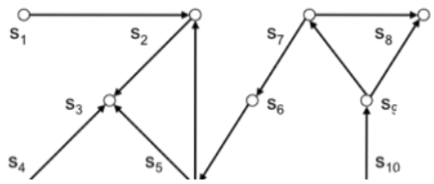
$CC(s_1) = \{s_1, s_2, s_3, s_4, s_5\}$

$CC(s_6) = \{s_6\}$

$CC(s_7) = \{s_7, s_8, s_9, s_{10}\}$



- Cet exemple présente un graphe orienté connexe :



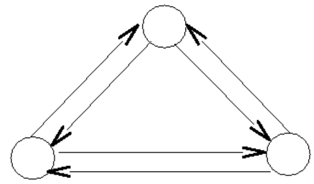
Remarques 1
- Un graphe ne possédant qu'une seule composante connexe est simplement un graphe connexe.
-Un sommet isolé (de degré 0) constitue toujours une composante connexe à lui seul.

Proposition 1 Soit G Un graphe, l'ajout d'une arête a pour conséquence :
– soit diminuer le nombre de composantes connexes.
– soit créer un nouveau cycle dans le graphe.

Proposition 2 Un graphe G d'ordre n connexe comporte au moins (n – 1) arêtes.

6.3 Graphe k-connexe

Un graphe G est dit **k-connexe** si et seulement si G est connexe d'ordre $n \geq k + 1$ et il n'existe pas d'ensemble $S \subset X$ de cardinal $k - 1$ tel que le sous graphe engendré par $X - S$ n'est pas connexe. En d'autres termes, en supprimant moins de k sommets, le graphe sera toujours connexe.
G est ***k*-sommet-connexe** si et seulement si le graphe reste fortement connexe, c'est-à-dire 1-arc-connexe, quand on luiôte $k - 1$ sommets.
Exemple: Ce graphe orienté complet à 3 sommets (Le graphe ci-contre) est 2-sommet-connexe.



Remarque
On définit la connectivité d'un graphe comme étant le plus grand k tel que G est k-connexe.

6.4 Forte Connexité

Définition 3 Un graphe **orienté** G=(X, U) est fortement connexe (f.c.) s'il existe entre chaque paire de sommets x et y ∈ X (x ≠ y) un chemin de x à y et un chemin de y à x .

6.5 Composantes fortement connexes (C.F.C.)

Soit G=(X, U) un graphe orienté :
Si le sous graphe engendré par un ensemble de sommets S ⊂ X (G_S) est fortement connexe et le sous graphe engendré par S ∪ x et x ∉ S n'est pas fortement connexe Alors G_S est une composante fortement connexe de G.

Application

Lors de la mise en place d'un plan de circulation dans une zone urbaine, il est intéressant de savoir s'il permet un déplacement de n'importe quel point à un autre. Un moyen de le vérifier est de représenter le plan, sous forme de graphe, et de vérifier qu'il est fortement connexe.

6.5.1 Algorithmes de calcul des composantes fortement connexes

Soit $G = (X,U)$ un graphe orienté, On définit pour chaque sommet $x \in X$ deux ensembles :

– L'ensemble des descendants de x : $D(x) = \{y \in X/x\ \boldsymbol{\sigma}\ y\}$
– L'ensemble des ascendants de x : $A(x) = \{y \in X/y\ \boldsymbol{\sigma} x\}$

Principe : A partir d'un sommet *r*, on calcule la c.f.c. à laquelle appartient *r*.

– On calcule l'ensemble des descendants de *r* (noté *D*).
– On calcule l'ensemble des ascendants de *r* (noté *A*).
– c.f.c. est $\{r\} \cup A \cap D$.

Données :

En entrée :

– *X* : ensemble de *n* éléments représentant les identificateurs des sommets.
– *U* : ensemble de *m* éléments sous forme (i, j) représentant les arcs où $i, j \in X$
– *r* : identificateur d'un sommet de *X* qu'on appelle racine.

En sortie :

– CFC : sous-ensemble de *X*. Le sous graphe engendré par cet ensemble représente une composante fortement connexe.

Autres :

– *i, j* : variables sommets.

– Marque : vecteur de n éléments booléens.
–*A*, *D* : sous-ensembles de *X*. Ils représentent l'ensemble des ascendants et l'ensemble des descendants de la racine.

Formulation 1 de L'algorithme :

(* Initialisation *)
$D \leftarrow \{r\}$;
Pour tout $i \in X$ *Marque*[*i*] ← *faux* ;
(* Recherche des descendants de r *)
Tant que ($\exists i \in D$) et (*Marque*[*i*] = *faux*)
    Marque[*i*] ← *vrai* ;
    Pour tout $(i, j) \in U$ $D \leftarrow D \cup \{j\}$;
(* Reinitialisation *)
$A \leftarrow \{r\}$;
Pour tout $i \in X$ *Marque*[*i*] à *faux* ;
(* Recherche des ascendants de r *)

Tant que $(\exists i \in A)$ et $(Marque[i] = faux)$

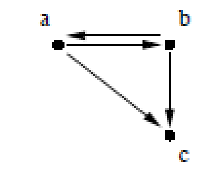
$Marque[i] \leftarrow vrai$

Pour tout $(j, i) \in U$ $A \leftarrow A \cup \{j\}$;

(* Calcul de la C.F.C. *)

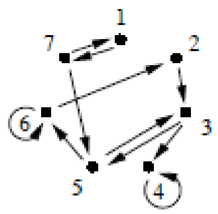
$CFC \leftarrow D \cap A$;

Exemple :



Exemples:

Ce graphe a deux composantes fortement connexes {a,b} et {c}



Ce graphe a trois composantes fortement connexes {1,7}, {2 ,3,5,6}, et {4}

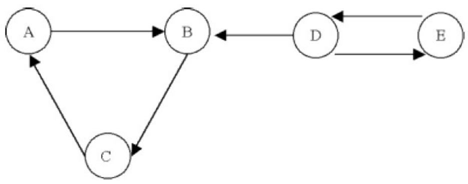
6.6 Graphe réduit

A tout graphe orienté $G = (X,U)$ on associe le graphe simple $GR = (X_R,U_R)$ appelé graphe réduit de G défini comme suit :

X_R = {A chaque c.f.c. Ci de G correspond un sommet de Ci}

$U_R = \{(Ci,Cj)/i \neq j$ et $\exists x \in Ci$ et $\exists y \in Cj$ et $(x, y) \in U\}$

Exemple



Ce graphe G n'est pas fortement connexe, car on ne peut pas trouver de chemin de A à D par exemple. Par contre, on identifie deux composantes fortement connexes A' = {A,B,C}et B' = {D,E}.

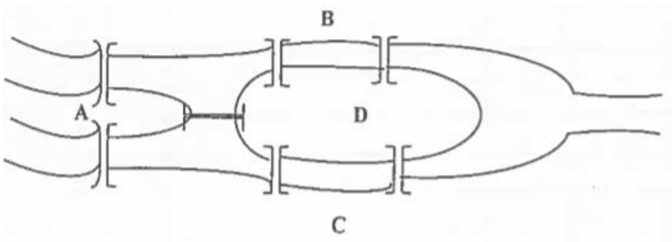
Le graphe réduit GR du graphe G est:

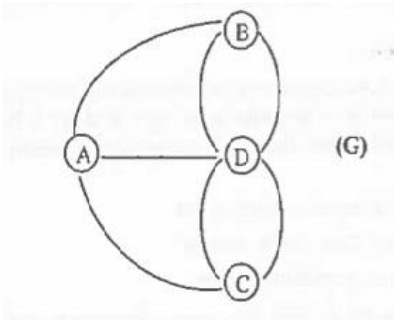
Remarque 6

- Un graphe fortement connexe possède une seule C.F.C.
- Le graphe réduit d'un graphe ne possède pas de circuits.

7. Parcours Eulériens

La ville de Koenigsberg possède deux iles (représentées par les régions A et B dans la figure 1.18(a)) et sept ponts. Le problème revient à trouver un cycle qui passe une seule fois par chacun de ses ponts et revient à son point de départ. Euler prouva que ce problème n'avait pas de solution et trouva une généralisation du problème.





(a) **(b)**

Figure 1.18 Les ponts de Koenigsberg (a) et le graphe correspondant (b).

Définition 1
Un parcours Eulérien passe une fois et une seule fois par chaque arête (resp. arc) du graphe. Le parcours peut être une chaine, un chemin, un cycle ou un circuit. Soit G un graphe contenant m arêtes (resp. m arcs) : Une chaine simple, un chemin simple, un cycle simple ou un circuit simple de longueur m est appelé Eulérien.

7.1 Chaine Eulérienne

Une chaine Eulérienne est une chaine qui passe par toutes les arêtes du graphe exactement une fois. Si la chaine est fermée, il s'agit d'un cycle Eulérien.

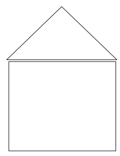
7.2 Chemin Eulérien

Un chemin Eulérien est un chemin qui passe par tous les arcs du graphe exactement une fois. Si le chemin est fermé, il s'agit d'un circuit Eulérien.

7.3 Graphe Eulérien

Un graphe connexe est dit graphe Eulérien si et seulement si il admet un cycle eulérien.

Exemple
Il s'agit d'une généralisation du jeu bien connu consistant à dessiner toutes les arêtes d'un graphe avec un crayon sans jamais le soulever, ni passer deux fois sur la même arête. Le graphe ci-dessous admet une chaine eulérienne mais il est **non eulérien** car il n'admet pas un cycle eulérien.



Théorème 1 *Euler 1766*
Un multigraphe G admet une chaine Eulérienne si et seulement si il est connexe (à des points isoles près) et le nombre de sommets de degré impair est 0 ou 2.
Théorème 2:
Un graphe connexe est eulérien si et seulement si tous ses sommets ont un degré pair.

Le graphe de la Figure 1.18(b) n'est pas eulérien car les sommets A,B,C et D ont un degréimpair.

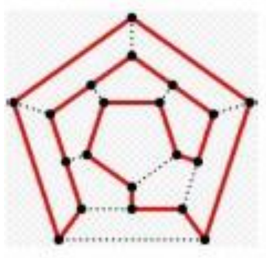
Conséquences
- Un graphe G admet une chaine Eulérienne d'un sommet x a un sommet y ($x \neq y$) si et

seulement si $dG(x)$ et $dG(y)$ sont impairs et $\forall z$ sommet de G ($z \neq x$ et $z \neq y$), on a $dG(z)$ pair.
- Un graphe G admet un cycle Eulerien si et seulement $\forall x$ sommet de G, on a $dG(x)$ pair.

Proposition 1
Un graphe $G = (X,U)$ admet un circuit Eulérien si et seulement si pour tout sommet x, on a $d^+G(x) = d^- G(x)$.

8. Parcours Hamiltoniens

Le problème Hamiltonien, inventé par W. Hamilton, revient à déterminer pour un ensemble de villes un itinéraire permettant de passer une et une seule fois par chaque ville et revenir au point de départ. La Figure 1.19 illustre ce type de problème.



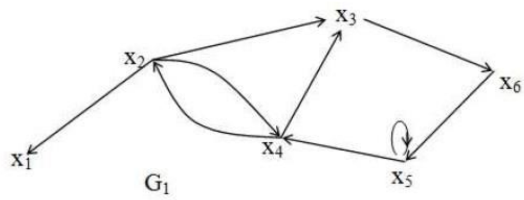


Figure 1.19 Représentation du problème Hamiltonien. **Figure 1.20** Graphe G1.

Définition 1
Un parcours Hamiltonien passe une fois et une seule fois par chaque sommet du graphe. Le parcours peut être une chaine, un chemin, un cycle ou un circuit. Soit G un graphe d'ordre n (contenant n sommets) : Une chaine (resp. un chemin) elementaire de longueur $n - 1$ est appelé chaine Hamiltonienne (resp. chemin Hamiltonien). Un cycle (resp. circuit) élémentaire de longueur n est appele cycle (resp. circuit) Hamiltonien.

8.1 Chaine Hamiltonienne

Une chaine Hamiltonienne est une chaine qui visite chaque sommet du graphe exactement une fois. Un cycle Hamiltonien est donc une chaine Hamiltonienne fermée.
Si on considère le graphe G1 de la Figure 1.20, on peut voir que la séquence (x4, x5, x6, x3,x2, x1) représente une chaine Hamiltonienne.

8.2 Chemin Hamiltonien

Un chemin Hamiltonien est un chemin qui visite chaque sommet du graphe exactement une fois. Un circuit Hamiltonien est donc un chemin Hamiltonien fermé.
Si on considère le graphe G1 de la Figure 1.20, on peut voir que la séquence (x3, x6, x5, x4,x2, x1) représente un chemin Hamiltonien.

8.3 Graphe Hamiltonien

Un graphe Hamiltonien est un graphe qui admet un cycle Hamiltonien. Le graphe G1 de la figure 1.20 ne comporte pas de cycle Hamiltonien car il est impossible de revenir à x1 partant de x1. Par conséquent, ce graphe n'est pas Hamiltonien.

Remarque:

Contrairement aux graphes eulériens, il n'existe pas de caractérisation simple des graphes hamiltoniens. On peut cependant énoncer quelques propriétés et conditions suffisantes.

1. Un graphe possédant un sommet de degré 1 ne peut être hamiltonien.
2. Si un sommet dans un graphe est de degré 2, alors les deux arêtes incidentes à ce sommet doivent faire partie du cycle hamiltonien.
3. Les graphes K_n sont hamiltoniens. (K_n étant un graphe complet de n sommets)

Théorème 1 (Ore) Soit G = (X,A) un graphe simple d'ordre n ≥ 3. Si pour toute paire de sommets {x, y} non adjacents, on a d(x) + d(y) > |X | alors G est hamiltonien.

Corollaire (Dirac) Soit G = (X,A) un graphe simple d'ordre n ≥3. Si pour tout sommet x de G on a d(x) ≥ n/2 alors G est hamiltonien.

Condition nécessaire

La contraposée du théorème suivant fournit une condition nécessaire pour avoir un graphe hamiltonien.

Théorème 1

Le théorème ci-dessous est utilisé pour démontrer qu'un graphe n'est pas Hamiltonien (ne contient pas de cycle Hamiltonien). Si $G = (X,E)$ est un graphe Hamiltonien, alors pour tout ensemble de sommets $S \subset X$, on a: $p(G_{X-S}) \leq |S|$ ou $p(G_{X-S})$ est le nombre de composantes connexes du sous graphe de G induit par l'ensemble $X - S$.

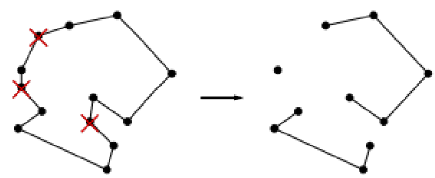
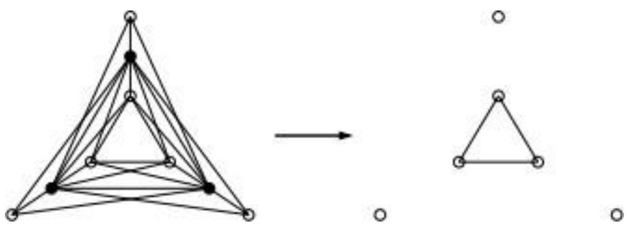


Figure 1.21 Graphe Hamiltonien ses composantes connexes après avoir retiré k sommets.

En d'autres termes, si on retire k sommets quelconques d'un graphe hamiltonien, on obtient auplus k composantes connexes. Voir Figure (1.21).

Exemple:

Le graphe suivant n'est pas hamiltonien car en enlevant les 3 sommets noirs, on obtient 4composantes connexes.



9. Coloration des Graphes

9.1 Coloration des sommets

La coloration des graphes (sommets) consiste à associer à chaque sommet une couleur de telle façon que chaque sommet adjacent n'est pas la même couleur.

9.1.1. Définition du nombre chromatique

Le nombre chromatique d'un graphe *G* est le nombre minimum de couleurs nécessaires à sa coloration, c'est-à-dire le plus petit nombre de couleurs permettant de colorier tous les sommets du graphe sans que deux sommets adjacents soient de la même couleur. On le note c.

Propriété 1
c <= d+1 où d: est le plus grand degré des sommets du graphe.
Propriété 2
n<= c où n: est l'ordre du sous graphe complet d'ordre le plus élevé. (la Clique maximale)
Propriété 3 Le nombre chromatique d'un graphe complet est égal à son ordre.

9.1.2. Algorithme de coloration

Un algorithme couramment utilisé est celui de **Welsh & Powell.** Il permet d'obtenir une assez bonne coloration d'un graphe, c'est-à-dire une coloration n'utilisant pas un trop grand nombre de couleurs. Cependant il n'assure pas que le nombre de couleurs utilisé soit minimum (et donc égal au nombre chromatique du graphe).

Algorithme de Welsh Pawell

Étape 1 Classer les sommets du graphe dans l'ordre décroissant de leur degré, et attribuer à chacun des sommets son numéro d'ordre dans la liste obtenue.
Étape 2 En parcourant la liste dans l'ordre, attribuer une couleur non encore utilisée au premier sommet non encore coloré, et attribuer cette même couleur à chaque sommet non encore coloré et non adjacent à un sommet de cette couleur.
Étape 3 S'il reste des sommets non colorés dans le graphe, revenir à l'étape 2. Sinon, la coloration est terminée.

9.1.3. Notion de clique/stable

- Un stable est un sous-graphe sans arête.
Définition: On appelle stable dans un graphe *G* un sous-ensemble de sommets $S \subseteq X$ tel que le sous-graphe engendré par *S* est formé de sommets isolés (ne contient aucun arc ou arête).

Dans le graphe de la Figure 1.22 ci contre :
L'ensemble de sommet $s = \{x1, x3\}$ est un stable. L'ensemble de sommet $s' = \{x2, x4\}$ est un autre stable.

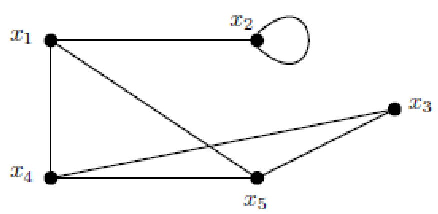


Figure 1.22 Exemple de graphe

- Une clique est un sous-graphe complet.

Définition: On appelle clique dans un graphe *G* un sous-ensemble de sommets $C \subseteq X$ ou le sous graphe engendre par *C* est un graphe complet.
L'ensemble de sommet $\{x1, x4, x5\}$ est une clique.

Remarque
- Chaque partition d'un graphe biparti est un stable.
- Si *S* est le plus grand stable dans *G* Alors $|S|$<= nombre de cliques dans *G* et

$|S|$ =nombre minimal de cliques dans G.

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

- Le **nombre chromatique** du graphe G, est le plus petit entier k pour lequel il existe une partition des sommets en k sous-ensembles stables.

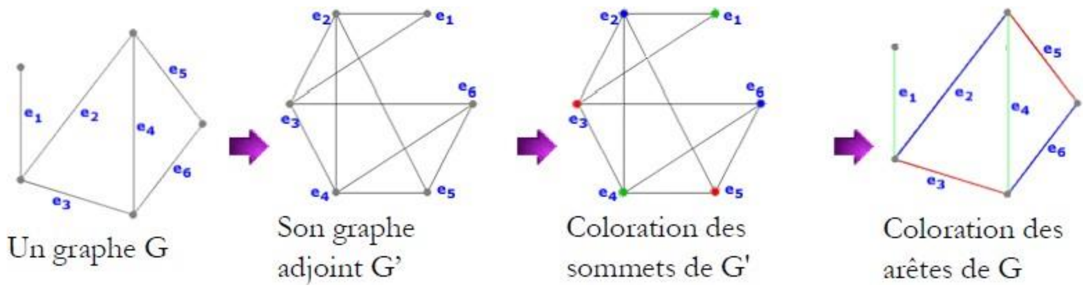
9.2. La coloration des arêtes

La coloration des arêtes d'un graphe consiste à affecter à toutes les arêtes de ce graphe une couleur de telle sorte que deux arêtes adjacentes ne portent pas la même couleur. L'indice chromatique du graphe G est le plus petit entier k pour lequel il existe une coloration des arêtes.

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

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

On peut ensuite appliquer par exemple l'algorithme de Welsh et Powell sur le graphe G' pour colorer ses sommets. Une fois cela fait, on colorera les arêtes de G de la même couleur que les sommets correspondants de G'. Exemple de coloration d'arêtes.



Chapitre 02
Algorithmes de Base en Théorie des Graphes

Objectifs:

Présentation des différents algorithmes de bases dans la théorie des graphes:

- Comprendre la notion d'arbre et la recherche dans une arborescence;
- Maîtriser l'algorithme de la recherche du plus court chemin;
- Maîtriser l'algorithme de la recherche du flot maximun;
- Maîtriser l'algorithme de la recherche d'un arbre couvrant de poids minimum.

Dans ce cours Vous allez voir quatre parties:

- **Partie 01:** Arbres et Arboresences (définitions),
- **Partie 02:** Le problème de recherche d'un plut court chemin,
- **Partie 03:** Le problème de la recherche du flot maximum.
- **Partie 04:** Le Problème de recherche d'un arbre de poids minimum,

Partie 01: Arbre et arboressance

1. Introduction

Les arbres sont des graphes particuliers, très utilisés en informatique globalement et en algorithmiques spécialement.

2. Construction d'un arbre

Définition 1:
Un arbre est un graphe connexe sans cycle.

Exemples :
La Figure 3.1 illustre un exemple de graphe représentant unarbre.

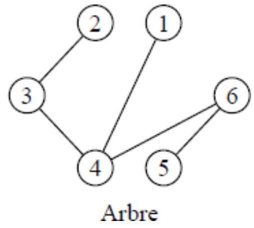


Figure 2.1 Graphe représentant un arbre.

Il existe de nombreuses caractérisations pour les arbres :

– En considérant la propriété de connexité, les arbres apparaissent comme des graphes connexes minimaux, dans le sens ou retirer la moindre arête les déconnecte.
– En considérant les arbres sous l'angle des graphes acycliques (sans cycle), ils sont maximaux, dans le sens ou ajouter la moindre arête crée un cycle.

Il existe également une relation très forte entre le nombre d'arêtes et le nombre de sommets d'un arbre : un arbre T=(V,E) comporte exactement $|V| - 1$ arêtes. En effet :

– Un graphe connexe comporte au moins $|V| - 1$ arêtes.
– Un graphe sans cycle comporte au plus $|V| - 1$ arêtes.

Théorème 1 : Théorème de caractérisation des arbres

Etant donné un graphe T=(V, E) d'ordre *n*. Les propriétés suivantes sont équivalentes et caractérisent un arbre.

(1) T est connexe et sans cycles;
(2) T est sans cycles et admet *n-1* arêtes;
(3) T est connexe et admet *n-1* arêtes;
(4) T est sans cycles et l'ajout d'une arête crée un cycle unique (le rend cyclique);
(5) T est connexe et la suppression d'une arête déconnecte T;
(6) Tout couple de sommets est relié par une chaine et une seule.

Preuve. Compte tenu de la définition d'un arbre, il suffit de montrer l'équivalence des propriétés (2) et (3).
–Supposons qu'un graphe connexe G à $n - 1$ arêtes possède un cycle. La suppression d'une arête de ce cycle créerai un graphe à $n - 2$ arêtes toujours connexe, ce qui est absurde. G est donc aussi acyclique.

– Supposons qu'un graphe acyclique G à $n - 1$ arêtes ne soit pas connexe. Il existerait alors deux sommets x et y qui ne peuvent être reliés par un chemin. Rajoutons alors une arête entre ces deux sommets. Le graphe obtenu est toujours acyclique mais possède maintenant n arêtes, ce qui est absurde. G est donc aussi connexe.

Définition 2

Une forêt est un graphe dont chaque composante connexe est un arbre.
Une forêt étant un graphe sans cycle, chacune de ses composantes connexes est acyclique, et par définition connexe. La définition d'une forêt correspond donc bien au sens usuel d'un ensemble d'arbres, chacune de ses composantes connexes étant un arbre.

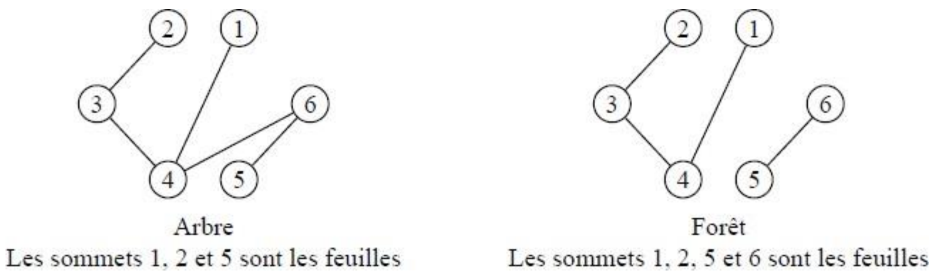


Figure 2.2 Arbre et forêt.

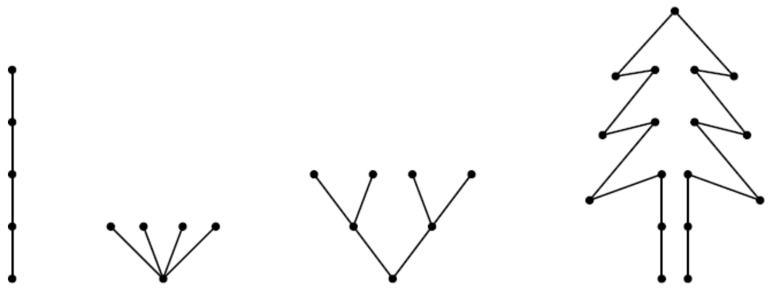


Figure 2.3 Quatre arbres, dont l'union est une forêt.

3. Arborescence

En informatique, plusieurs applications nécessitent l'utilisation d'arborescences, particulièrement pour les arbres binaires. Comme exemple d'applications les arbres généalogiques.

Définition 3
Si un arbre possède un sommet particulier r qui peut mener vers tous les autres sommets ($\forall$ x $\in$ X , il existe un chemin de r vers x), alors cet arbre est qualifié d'arborescence.

Une arborescence est un graphe G =(V,E) tel que :
- r $\in$ X est la racine de G.
- G est un arbre.

Remarques :
- Une arborescence est donc un arbre mais l'inverse n'est pas toujours vrai.
- Le concept d'arborescence est orienté puisque la notion de racine et de chemin est utilisée dans sa définition (même si ce n'est pas mentionné sur le graphe).

3 .1 Caractérisation des arborescences

Théorème 4 : Théorème de caractérisation des arborescences

Etant donné un graphe T=(V,E) d'ordre n $\geq$2, les conditions suivantes sont équivalentes et caractérisent une arborescence :

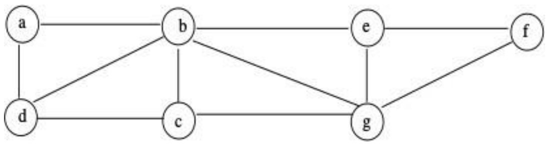
(1) T est un arbre admettant le sommet r pour racine;
(2) $\forall$ x $\in$ X il existe un chemin unique de r vers x;
(3) T admet r pour racine et est minimal pour cette propriété (si on supprime un arc, r n'est plus racine);
(4) T est connexe et $d^-(r) = 0$ et $\forall$ x $\in$ X $d^-(x) = 1$;
(5) T est sans cycles et $d^-(r) = 0$ et $\forall$ x $\in$ X $d^-(x) = 1$;
(6) T admet r comme racine et est sans cycles;
(7) T admet r comme racine et possède n-1 arcs;

La racine ne possède pas de prédécesseurs, les arcs sont tous orientés de la racine vers les feuilles. Les feuilles sont les sommets n'ayant pas de successeurs.

Remarque: Une arboressance est un arbre mais la réciproque est fausse.

Exemples:

Exemple 01: On considère le graphe non orienté suivant:

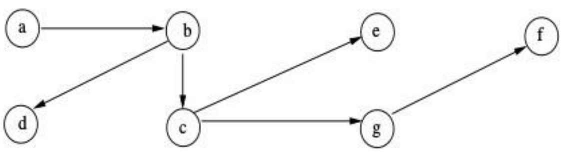


Combien faut-il enlever d'arêtes de ce graphe pour le transformer en arbre ? Donnez un graphepartiel de ce graphe qui soit un arbre.

Correction: le graphe comporte 7 sommets et 11 arêtes. Pour le transformer en arbre il faudra donc enlever 5 arêtes. Par exemple, les arêtes (f, g), (b, g), (b, c), (b, d) et (a, d).

Exemple 02: On considère le graphe précédent. Est-il possible d'extraire une arboressance dont la racine est le sommet a?

Correction: Oui, on peut car le sommet a menra vers tout les autres sommets.
Voilà l'aborressance:



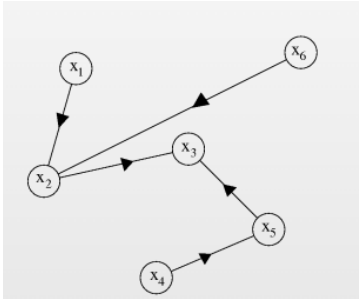
3. Anti- arborescence

Etant donné un graphe T=(V, E) d'ordre n $\geq$2, les conditions suivantes sont équivalentes et caractérisent une arborescence :

(1) G est un arbre;
(2) $\forall$ x $\in$ X il existe un chemin unique de x vers r.

(3) r est une anti-racine de G.

Exemple: La figure suivante représente une anti-arboressance avec x_2 comme anti-racine.



Remarque:
Si on inverse le sens des arcs d'une arboressance, on obtient une anti-arboressance.

Partie 02: Problème du plus court chemin

1. Introduction

La recherche d'un plus "court" chemin d'un point à un autre est un problème de la vie courante. On s'intéresse dans cette partie à la résolution de ce type de problème. On considère des graphes orientés valués, tels qu'une valeur est associée à chaque arc, et l'on cherche le plus court chemin entre deux sommets du graphe. Cela permettra de résoudre des problèmes comme la recherche d'un itinéraire coûtant le moins cher, ou encore étant le plus rapide.
Il existe de nombreux problèmes spécifiques aux chemins. Citons quelques uns:

1) Soient deux sommets i et j $\in$ X:
- Existe-t-il au moins un chemin de i à j ?
- Existe-t-il au moins un chemin de longueur p entre i et j ?
- Quel est le nombre de chemins de longueur inférieure ou égale à p entre i et j ?

2) Trouver le ou les chemins qui relient le sommet s à tous les autres sommets et qui satisfont une propriété P.

3) Trouver les chemins qui relient tout couple de sommets et qui satisfont une propriété P.

Le troisième problème est le plus simple à résoudre : l'algorithme de Floyd-Warshall nous en donnera une solution. En revanche et de manière surprenante il n'existe pas à l'heure actuelle d'algorithme qui donne la solution du premier problème sans résoudre le second. Nous donnerons une solution de ces deux problèmes en étudiant l'algorithme de Dijkstra. Il faut cependant noter qu'il existe de multiples algorithmes de plus courts chemins, souvent adaptés à un type particulier de graphe.

2. Introduction au problème du plus court chemin

Etant donné un graphe G=(X,A) orienté tel que chaque arc (i,j) possède un nombre α (i,j) appelé longueur de l'arc (i,j). Le problème du plus court chemin (PCC) consiste à déterminer un chemin élémentaire C_h joignant les sommets i et j et qui soit de longueur extrémale (minimale ou maximale selon l'objectif). La longueur du chemin optimal est égale à la somme des longueurs des arcs du chemin.

On considère d'une manière plus générale deux types de problèmes :

- **Problème P1:** Il s'agit de trouver le plus court chemin reliant un sommet à tous les autres.
- **Problème P2:** il s'agit de trouver tous les plus courts chemins entre tous les couples de sommets.

La Figure 2.4 représente un réseau routier à deux extrémités (ou réseau de transport) comportant un ensemble de sommets X consistant en une entrée S, une sortie P et trois autres sommets. Chaque arc du réseau est assigné d'une valeur représentant sa longueur.

3. Algorithmes proposés pour trouver le PCC

Pour la résolution du problème du plus court chemin, divers algorithmes de complexité en temps polynomial ont été proposés. Leur application dépend souvent de la nature du graphe. Si l'on considère un problème de type P1, on a le choix entre les algorithmes de Dijkstra et de Bellman–Ford.

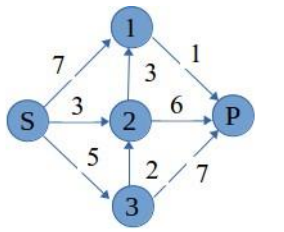


Figure 2.4 Exemple d'un réseau routier.

Lorsque **le graphe ne comporte pas de cycle**, on utilise généralement l'algorithme de **Bellman–Ford**. Lorsque **les longueurs des arcs sont toutes positives**, on utilise l'algorithme de **Djikstra**.

Pour les problèmes de type P2, on a également le choix entre deux algorithmes: l'algorithme de Dantzig et celui de Floyd-Warshall qui sont des algorithmes matriciels applicables quelque soient les longueurs des arcs. Dans ce cours, on s'intéresse aux problème de type p1.

3.1 Quelques notions fondamentales:

3.1.1 Graphe pondéré (Réseau)

Soit G=(X,U) un graphe orienté, On définit $p : U \to R$ une application qui associe à chaque arc $u \in U$ de G une valeur réelle $p(u)$ appelée poids de l'arc u. Un tel graphe, représenté par $G=(X,U,p)$, est appelé graphe pondéré, graphe valué ou réseau.

3.1.2 Poids d'un chemin

On définit le poids d'un chemin *ch* comme la somme des poids des arcs de *ch* , $p(u)=\sum_{u \in ch} p(u)$.

On l'appelle aussi distance.

3.1.3 Circuit absorbant

Un circuit est dit absorbant si son poids est négatif. $p(u)<0$

Si un graphe possède un circuit absorbant, alors il n'existe pas de plus courts chemins entre certains de ces sommets.

3.2 Résolution de problèmes de type P1

Dans cette section sont présentés les algorithmes suivants: l'algorithme de Dijkstra et l'algorithme de Bellman-Ford.

Les algorithmes de Dijkstra et Bellman-Ford procèdent tous les deux par relâchements successifs d'arcs. La différence entre les deux est que dans l'algorithme de Dijkstra, chaque arc est relâché une et une seule fois, tandis que dans l'algorithme de Bellman-Ford, chaque arc peut être relâché plusieurs fois.

3.2.1 Algorithme de Djikstra

EdgserWybe Dijkstra (1930-2002) a proposé en 1959 un algorithme qui permet de calculer le plus court chemin entre un sommet particulier et tous les autres. Le problème est le même que celui traité par Bellman-Ford, il utilise aussi le principe de la programmation dynamique, sauf qu'ici le réseau R=(X,U,p) peut être avec ou sans circuits mais à condition que les poids

soient tous positifs ou nuls. Donc, L'algorithme de Dijkstra ne permet pas de considérer les arcs négatifs, car une fois qu'un sommet est marqué on ne peut changer ce marquage lors des itérations suivantes. Le sommet de départ s n'est pas nécessairement une source.
Principe L'algorithme calcule le chemin de poids minimal en partant du sommet s et en prolongeant le chemin à chaque itération. Cette méthode s'appelle, calcul de la plus courte distance de proche en proche. A chaque étape, pour un sommet donné *x*, on ajuste les valeurs des poids L pour tout sommet y successeur de *x*.

Algorithme : Dijkstra (G,s)

Données: Un réseau R=(X, U, d) avec d(u)>= 0

Etape 1 : initialisation
Soit $s \in X$ */*sommet de départ*/*
Soit   L(x) : le label de x ou la longueur du pcc de s à x à l'etape i
$X_m \leftarrow \phi$; $L(s) \leftarrow 0$;   */* X_m est l'ensemble des sommets marqués*/*
$\forall x \in X - \{s\}$: $L(x) = \infty$;
$A_m \leftarrow \phi$;   */* A_m est l'ensemble des arcs du PCC */*

Etape 2 : **Traitement**
Tant que Card (X_m)< n Faire
        $x \leftarrow$ Min (L(x)), $\forall$ / $x \in X - X_m$ */*Choisir x de plus petit label parmi les sommets non marques **
        $X_m \leftarrow X_m \cup \{x\}$;   */*marquer x */*
        $A_m \leftarrow A_m \cup \{(pred(x), x)\}$;   */*marquer l'arc de provenance du min*/*
        Pour tout $y \in$ Voisins(x) et $y \notin X_m$ Faire
                L(y) = Min(L(y), L(x)+ d(x,y)) ;
        Fin pour
Fin tant que

Etape 3 : Résultat
A_m contient les arcs du PCC
L(x) est la longueur du PCC
Fin

Exemple: On donne en Figure 2.5 un graphe représentant un réseau. Appliquer l'algorithme de Dijkstra pour déterminer le PCC entre les sommets S et P.

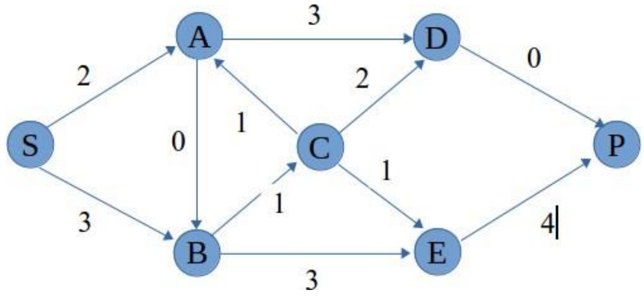


Figure 2.5 Plus court chemin de S à P de longueur 5.

Sommet / Etapes	S	A	B	C	D	E	P	**Sommet fixé**	**L_{min}**
Initi.	0	∞	∞	∞	∞	∞	∞	**S**	**0**
1		2S	3S	∞	∞	∞	∞	**A**	**2**
2			2A	∞	5A	∞	∞	**B**	**2**
3				3B	5A	5B	∞	**C**	**3**
4					5A	4C	∞	**E**	**4**
5					5A	-	8E	**D**	**5**
6							5D	**P**	**5**

Pour retrouver le PCC, on part du sommet P →D→ A→S;

Ce chemin est lu depuis le tableau de la façon suivante : le dernier sommet fixé est le p, et depuis la colonne P, le dernier sommet utilisé dans cette colonne est le sommet D. On visite la colonne du sommet D et on trouve qu'il vient du sommet A et finalement, on visite la colonne A pour trouver que le sommet A vient du sommet S.

Donc, si on commence du S :
- Le PCC de S à P est : S-A-D-P
- Sa valeur = 5

Remarque : Les valeurs du tableau sont calculer en utilisant la phase de traitement (étape 2) mentionné dans l'algorithme.

3.2.2 Algorithme de Bellman-Ford

L'algorithme de Bellman-Ford utilise le principe de la programmation dynamique pour déterminer le plus court chemin dans un graphe orienté pondéré. Il est applicable même dans le cas où certains arcs ont une longueur négative. La seule contrainte est que le graphe ne doit pas comporter de circuits absorbant. L'algorithme de Bellman-Ford permet un marquage des sommets qui n'est pas définitif et qui peut être modifié à chaque itération. Ce type d'algorithme est dit à correction d'étiquettes.
A chaque sommet $x \in X$, on veut associer un chemin de poids optimal joignant la source du graphe $r \in X$ a x dans le réseau $R = (X,U, p)$

Principe L'algorithme de Bellman-Ford fonctionne selon le même principe que celui de Dijkstra : on associe à chaque sommet x_i une valeur $d[x_i]$ qui représente une borne maximale du coût du plus court chemin entre s_0 et x_i . L'algorithme diminue alors progressivement les valeurs $d[x_i]$ en relâchant les arcs. Contrairement à Dijkstra, chaque arc va être relâché plusieurs fois. On relâche une première fois tous les arcs ; après quoi, tous les plus courts chemins de longueur 1, partant de s, auront été trouvés. On relâche alors une deuxième fois tous les arcs ; après quoi tous les plus courts chemins de longueur 2, partant de s, auront été trouvés... et ainsi de suite... Après la k-ième série de relâchement des arcs, tous les plus courts chemins de longueur k, partant de s, auront été trouvés. Étant donné que le graphe ne comporte pas de circuit absorbant, un plus court chemin est nécessairement élémentaire.

Algorithme: Bellman-Ford (G,s)
Données: Un réseau R=(X, U, d) avec d(u)>= 0

Déclaration des variable
Graphe (S,A)
Soit s ∈ X /*sommet de départ*/
L_{uv} = ∞ /* S'il n'y a pas d'arc entre u et v */

Etape 1 : initialisation

d(s) ← 0
Pour chaque v ∈ S sauf (s) Faire
d(v) ← ∞

Etape 2 : Traitement (Phase de relaxation)

Pour i ← 1 à S-1 faire
Pour chaque arc (u,v) ∈ A Faire /* Pour tous les arcs : min (d(v, d(u)+C(u,v)))*/
Si d(v) >d(u) +C(u,v) alors
d(v)← d(u)+ C(u, v)
Fin
Fin

Etape 3 : Résultat (Phase de Contrôle de la présence d'une boucle négative)

Pour chaque arc (u ,v) ∈ A Faire
Si d(v) > d(u)+ L(u,v) Alors
Existence d'une boucle négative
Sinon retourner d(v)

Fin

Exemple : Etant donné le graphe de la Figure 2.6, appliquer l'algorithme de Bellman-Ford pourdéterminer le PCC entre les sommets S et P.

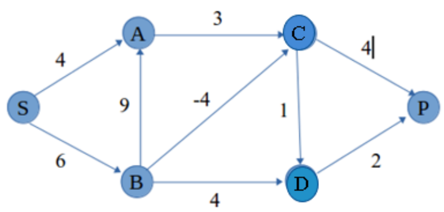


Figure 2.6 Plus court chemin de S à P de longueur 5.

Arc	poids
S-A	4
S-B	6
A-C	3
B-A	9
B-C	-4
B-D	4
C-D	1
C-P	4
D-P	2

Sommet itération	S	A	B	C	D	P
Initialisation	0		∞	∞	∞	∞
1	0	4S	6S	∞	∞	∞
2	0	4S ~~15B~~	6S	~~7A~~ 2B	10B	6C
3	0	4S	6S	2B	3C	~~6C~~ 12 D
4	0	4S	6S	2B	3C	~~6C~~ 5D
5	0	4S	6S	2B	3C	5D

Pour retrouver le PCC, on part du sommet P →D→C→B→S
D'après le tableau de résolution, on conclut que:
- Le PCC de S à P est : S-B-C-D-P
- Sa valeur = 5

Le chemin trouvé est lu de la même façon que l'algorithme précédent et es valeurs du tableau sont calculer en utilisant la phase de traitement (étape 2) mentionné dans l'algorithme.

Remarques

- Si le chemin optimal est le chemin de poids maximal, on change dans l'algorithme min par max et +∞ par −∞.
- Si $\forall x \in X$, $p(x) = 1$, l'algorithme calculera le plus court chemin en nombre d'arcs.
- Dans le cas où il n'y a pas de circuits absorbants :

1. l'algorithme se termine (les valeurs se stabilisent) après au plus *n* passages dans la boucle principale, *n* étant le nombre de sommets du graphe
2. Les valeurs L(x) obtenues à la fin de l'algorithme sont bien les distances des x au sommet s.

- L'algorithme permet de détecter la présence de circuits absorbants (un cycle de poids négatifs) : si les valeurs L(x) ne sont pas stabilisées après *n* passages de boucles (un nouveau tour de boucle ferait diminuer une distance), alors le graphe contient au moins un circuit absorbant.

3.3 Résumé

Les algorithmes de résolution du PCC

Algorithme	Type du PCC	Type du graphe	Avantages	Inconvénients
Dijkstra	D'un sommet à tous les autres sommets	Orienté ou non orienté	Un temps d'exécution assez rapide	• Ne s'applique qu'aux graphes à valuations positives • Ne marche que pour trouver les plus courts chemins.
Bellman		Graphe orienté sans circuit	Longueur quelconque (nombre réel) Un temps d'exécution assez rapide	- Ne s'applique qu'aux graphes décomposables en niveaux (donc sans circuits, sommet d'origine doit être sans prédécesseur).
Bellman Ford		Graphe orienté	Fonctionne sur tous les graphes (Longueur d'arc quelconque, nombre réel) consomme moins de mémoire.	- Un temps de calcul encore relativement long .
Floyd-Warshell	Entre tous les couples de sommets	Graphe orienté et sans circuit	Fonctionne sur tous les graphes	- Temps de calcul est très s long. - Consomme beaucoup de mémoire. - Calcule l'ensemble des distances entre tout couple de sommets alors qu'on peut souvent se contenter des distances depuis un sommet particulier.

Partie 03: Flots (Flots Maximum)

1. Introduction

Dans cette partie du chapitre, seront examinés des réseaux dans lesquels les sommets sont reliés par des arcs à travers lesquels une certaine quantité homogène peut s'écouler. Cette quantité est appelée flot.

Les flots permettent de modéliser une très large classe de problèmes. Leur interprétation correspond à la circulation de flux physiques sur un réseau: distribution électrique, réseau d'adduction d'eau ou tout autre liquide, acheminement de paquets sur un réseau informatique, ...

Il s'agit d'acheminer la plus grande quantité possible de matière entre une source s et une destination p. Les liens permettant d'acheminer les flux ont une capacité limitée, et il n'y a ni perte ni création de matière lors de l'acheminement: pour chaque nœud intermédiaire du réseau, le flux entrant (ce qui arrive) doit être égal au flux sortant (ce qui repart) (Conservation des flux en chaque sommet: loi de Kirchhoff).

2. Définitions et propriétés

2.1 Arcs incidents à un sommet

Dans ce qui suit, les arcs du graphe sont supposés numérotés A = (a1, a2, ….., ap).

Si l'on considère un ensemble Y de sommets tel que $Y \subset X$, on désigne par:

- ω^+ l'ensemble des arcs incidents à Y vers l'extérieur (ensemble des arcs d'origine x et d'extrémité différente de x)

$\omega^+(Y) = \{(x, y) \in A \,/\, x \in Y \text{ et } y \notin Y\}$

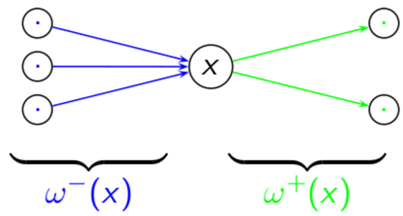


Figure 2.7 Arcs incidents à un sommet.

- ω^- l'ensemble des arcs incidents à Y vers l'intérieur (ensemble des arcs d'extrémité y et d'origine différente de y)

$\omega^-(Y) = \{(x, y) \in A \,/\, x \notin Y \text{ et } y \in Y\}$

Les arcs de $\omega^+(y)$ sont les arcs "sortant de y" et ceux de $\omega^-(y)$ les arcs "entrant en y".

Sur le graphe G1 de la Figure 2.8 , on a:

$\omega^+(a) = \{u1 , u5, \omega^-(a) = \{u4\}$, $\omega^+(b) = \{u2\}$, $\omega^-(b) = \{u1, u6\}$.

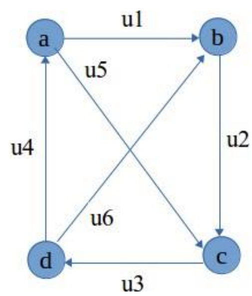


Figure 2.8 Le graphe G1.

2.2 Notion de flot

D'une manière générale, la notion de flot fait référence à une quantité homogène qui peut s'écouler à travers les arcs d'un graphe. Une définition plus formelle peut être donnée comme suit:

2.2.1 Définition d'un flot

Définition 1 : Etant donné un graphe G = (X, A), un flot sur G est un vecteur,

$$\varphi = (\varphi_1 , \varphi_2 , \varphi_3 , \dots \varphi_n) \in R^p \text{ tel que :}$$

- En tout sommet $x_i \in X$, la loi de Kirchoff est vérifiée (loi de conservation de flot aux nœuds) :

$$\Sigma i / a_i \in \omega^+ (x_i) \varphi_i = \sum_i / a_i \in \omega^- (x_i) \varphi_i$$

Exemple :

Dans le graphe de la Figure 2.9, nous avons représenté sur tous les arcs un flot , φ = (3, 3,-2, 2, 1).

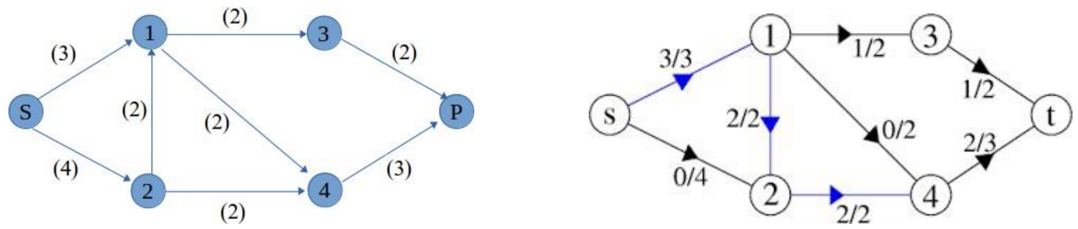


Figure 2.9 Représentation d'un flot dans un graphe. Le graphe à gauche représente la capacité maximale de chaque arc. Le graphe à droit représente le flux réelement passé sur chaque arc.

2.3 Réseau de transport

Définition 2:

Soit un graphe G=(X, A) orienté et sans boucles. G est un réseau de transport si :

1. G est connexe.
2. G admet deux sommets particuliers s et p appelés **source et puits :**
   - Une seule entrée S (sommet source) tel que
   $d^-(S) = 0$ et ∀ x ∈ X, il existe un chemin de S à x.
   - G admet une seule sortie P (sommet puits) tel que $d^+(P) = 0$.
   Une seule sortie P (sommet puits) tel que $d^+(P) = 0$.
3. Les arcs de G sont valués et la valeur de l'arc notée c(a) est la capacité de l'arc a. La fonction capacité c : A → R+ limite les valeurs que peut prendre la fonction de flot sur les arcs.

Exemple:

Dans la Figure 2.10, sont représentés deux graphes. Le graphe de la partie (a) est un réseau de transport car il répond à toutes les conditions spécifiées dans la définition. Le graphe de la partie (b) de la même figure n'est pas un réseau de transport car il contient deux sommets source.

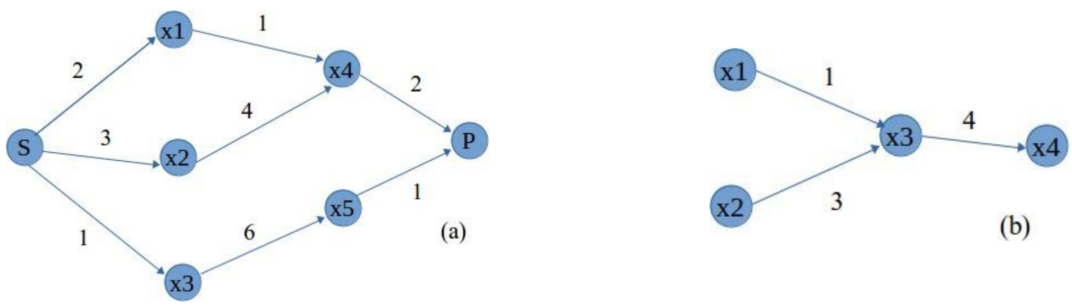
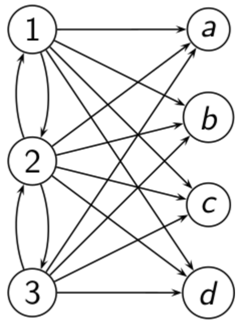
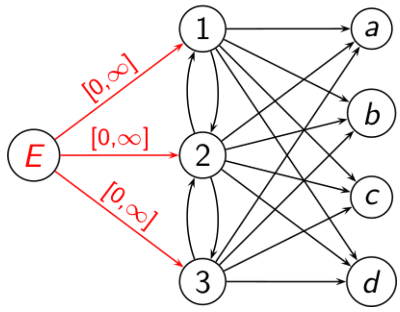


Figure 2.10 Le graphes en (a) est un réseau de transport et celui en (b) n'est pas un réseau.

Remarque 1:

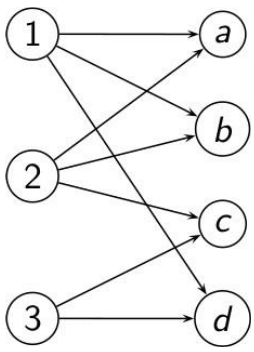
- Si G comporte plusieurs racines e_1 , e_2 , ..., e_n (le graphe à gauche) , on se ramène à un réseau en ajoutant à X un sommet E et à U (E , e_i), $\forall$ i = 1, . . . , n, avec $k(E , e_i) = +\infty$, $b(E , e_i) = 0$, $c(E , e_i) = 0$.

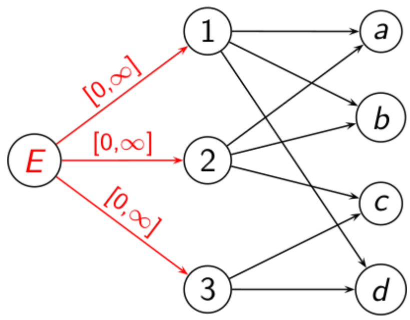




Remarque 2:

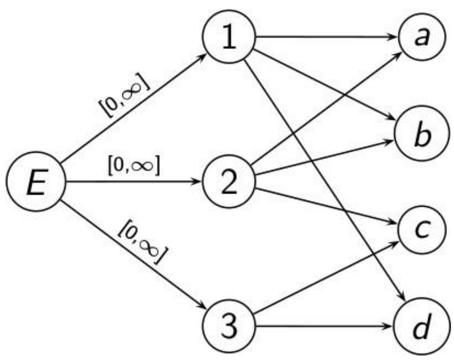
- Si G n'a pas de racine (le graphe à gauche), on ajoute un sommet E prédécesseur de tous les sommets x t.q. $\Gamma^{-1}(x) = \varnothing$, $k(E , x) = +\infty$, $b(E , x) = 0$, $c(E , x) = 0$.

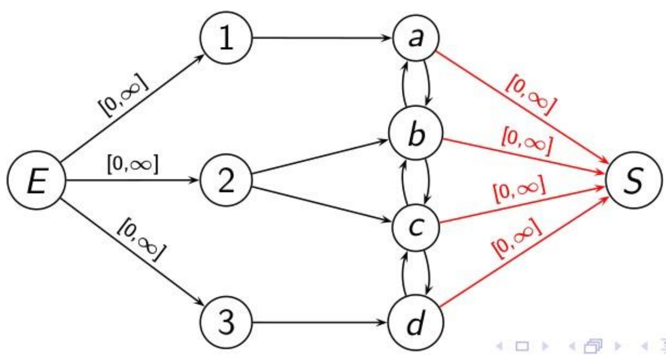




Remarque 3:

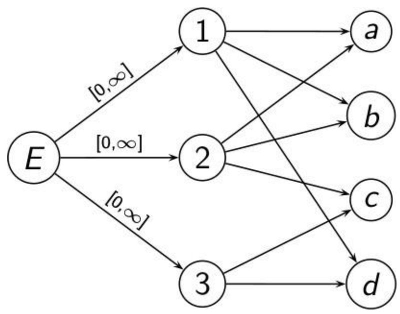
Si G comporte plusieurs antiracines s1 , s2 , ..., sn (le graphe à gauche), on se ramène à un réseau en ajoutant à X un sommet S et à U (s i , S), $\forall$ i = 1, . . . , n, avec $k(s_i , S) = +\infty$, $b(s_i , S) = 0$, $c(s_i , S) = 0$.

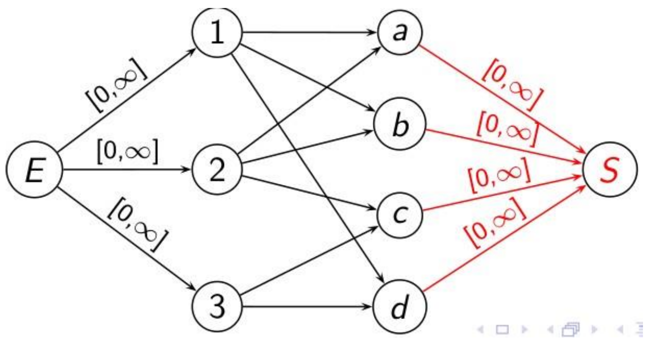




Remarque 4:

- Si G n'a pas d'antiracine, on ajoute un sommet S successeur à tous les sommets x t.q. $\Gamma(x) = \varnothing$, $k(x, S) = +\infty$, $b(x, S) = 0$, $c(x, S) = 0$.





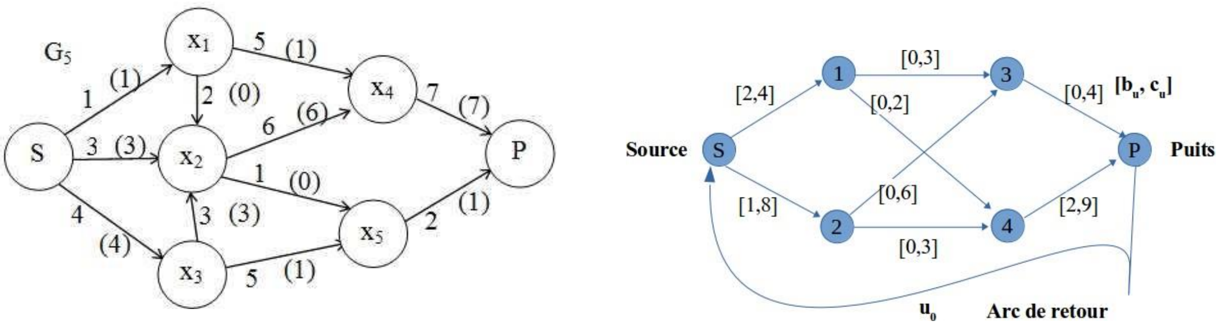
2.4 Flot dans un réseau de transport

Dans un réseau de transport, la notion de flot est légèrement affinée.
Définition 3 : Etant donné un graphe G = (X, A) constituant un réseau de transport. Un flot dans G est un vecteur , $\varphi = (\varphi_1, \varphi_2, \varphi_3, \dots \varphi_n) \in R^p$ tel que:

- $b_i \leq \varphi_i \leq c_i$, $\forall i \in \{1, 2, \dots\dots, p\}$
- En tout sommet $x_i \in X\text{-}\{S,P\}$, la loi de Kirchoff est vérifiée (loi de conservation de flot aux noeuds) : $\Sigma i/a_i \in \omega^+(x_i)\varphi_i = \sum_i/a_i \in \omega^-(x_i)\varphi_i$

Pour un réseau déterminé, tout vecteur de flot satisfaisant ces équations est un flot réalisable, ce qui signifie qu'il peut être réalisé dans le réseau.

Exemple :
La Figure 2.11 présente deux réseaux de transport avec sur tous les arcs une capacité et un flot.



(a) (b)

Figure 2.11 Le flot dans un réseau de transport. (a) les borne inf bi=0 (b) les bi $\neq$ 0.

Sur le graphe (a) :
- c = (1, 3, 4, 5, 2, 3, 5, 6, 1, 7, 2)
- φ = (1, 3, 4, 1, 0, 3, 1, 6, 0, 7, 1)

Définition 4 : La quantité de flot qui circule dans un réseau est calculée à l'entrée ou à la sortie du réseau:

$$\mathfrak{I}(xi) = \Sigma i/a_i \in \omega^+(x_i)\varphi_i = \sum_i/a_i \in \omega^-(x_i)\varphi_i$$

2.5 Flot complet

Dans un réseau de transport, on dit qu'un arc a $\in$A est saturé si , $\varphi(a) = c(a)$

Définition 5: Un flot φ, est dit complet si chaque chemin de S à P contient au moins un arc saturé.

Exemple :
Le flot présenté dans la Figure 2.7 est un flot complet. On peut facilement vérifier que chaquechemin de S vers P contient au minimum un arc saturé.

- c = (1, 3, 4, 5, 2, 3, 5, 6, 1, 7, 2)
- φ = (1, 3, 4, 1, 0, 3, 1, 6, 0, 7, 1)

2.6 Coupe

D'une manière générale, une coupe est un ensemble d'arcs dont la suppression sépare un graphe connexe en deux composantes connexes, l'une contenant S et l'autre contenant P.

2.6.1 Définition d'une coupe

Définition 6 :
Etant donné un réseau de transport, une coupe S-P, $\mathcal{T}$ notée = <X1, X2>
tel que : S∈ X_1 et P ∈ X_2. La capacité d'une telle coupe (égale à la somme des flots sur ses arcs) est :

$$C(\mathcal{T})=\sum_{\substack{i\in X_1\\ j\in X_2}} c(i,j)$$

Définition 7:
Soit s, un sous-ensemble de sommets, tel que S ∈ s et P ∉ S;
Une coupe séparant S et P est l'ensemble des arcs $\omega(w) = \omega^{+}(w) \cup \omega^{-}(w)$

$\omega^{+}(S)$ = {u∈U / origine de u dans S et extrémité hors de S}

$\omega^{-}(S)$ = {u∈U / extrémité de u dans S et origine hors de S}

La capacité de la coupe est égale à :

$$C(S)= \sum_{u\in\omega^{+}(s)} c_u + \sum_{u\in\omega^{-}(s)} b_u$$

Exemple:
Si l'on considère le réseau de transport de la Figure 2.12 .
On peut représenter 3 coupes en indiquant la liste des sommets dans cette coupe:
Coupe 1: $\mathcal{T}$ 1 = <(S, x1), (S, x2), (S, x3)>
Coupe 2: $\mathcal{T}$ 2 = <(x1, x4), (x2, x4), (x2, x5), (x3,x5)>
Coupe 3: $\mathcal{T}$ 3 = <(x4, P), (x5, P))>
De plus,

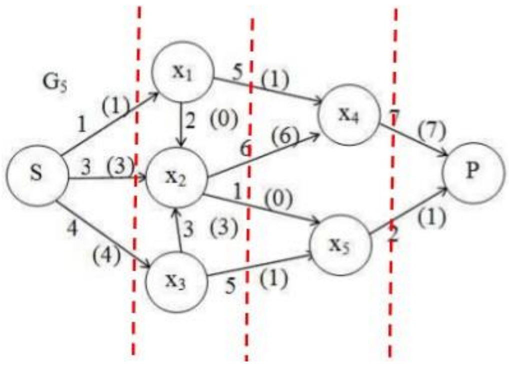


Figure 2.12 Représentation des différentes coupes dans un réseau de transport.

$\mathcal{T}$ (S_1) = 1+3+4 = 8

$\mathcal{T}$ (S_2)= 1+6+0+1 = 8

$\mathcal{T}$ (S_3) = 7+1=8

La capacité d'une coupe C = <X1, X2> est également notée c(X1, X2). Par ailleurs, leflot $\mathcal{T}$ d'une coupe = <X1, X2> est défini comme :

$$\varphi^{+}(C) = \sum_{\substack{i\in X_1\\ j\in X_2}} \varphi(i,j)$$

La valeur d'un flot peut à présent être obtenue partir des flots de n'importe quelle coupe C.

2.7 Coupe minimale

On s'intéresse à présent à une coupe particulière d'un réseau de transport. Il s'agit d'une coupe C telle que tous les arcs rentrant de C soient saturés et tous les arcs sortants de C, $\mathfrak{C}$ portent un flot nul. Cette coupe est appelée coupe minimale.

Définition 8 : Dans un réseau de transport, une coupe C est une coupe minimale si :

- L'arc (i, j) est saturé si $i \in X_1$ et $j \in X_2$
- $\varphi(i,j) = 0$ si $i \in X_2$ et $j \in X_1$

2.8 Définition du problème du flot maximal

Le problème du flot maximal est celui de la détermination d'un flot sur G, compatible avec les capacités, de telle façon qu'il soit le plus grand possible c'est à dire la quantité maximale du flot qu'on puisse le faire passer sur G.

3. Recherche du flot maximum

Il est fréquent de faire recours à la représentation d'un réseau de transport de marchandises ou un réseau électrique au moyen d'un graphe. Le problème de flot dans ces réseaux revient à étudier la circulation de matière continue ou discrète sur les arcs du graphe et qui obéit au principe de conservation de la matière.

Etant donné un réseau de transport, nous pouvons considérer deux problèmes. Le premier consiste en recherche de la valeur du flot réalisable maximum dans le réseau. Le second consiste à rechercher une fonction de flot satisfaisant cette valeur. Le premier est résolu par un théorème alors que le second est résolu par un algorithme qui permet de former une fonction de flot maximum. Le théorème et l'algorithme sont tous deux dus à Ford et Fulkerson.

3.1 Théorème de Ford-Fulkerson

Théorème 1 : (Théorème de Ford-Fulkerson) (Max-flow min-cut)
Etant donné G = (X, A) un réseau de transport et φ, un flot réalisable sur G.
La valeur du flot maximum réalisable de S vers P est égale à la capacité de la coupe minimale séparant S et P.

$$\text{Max}(\varphi) = \text{Min}(c(\mathfrak{C}))$$

D'après ce théorème, nous pouvons affirmer que s'il existe un flot φ, de valeur v, égale à la capacité d'une coupe (X1, X2) alors le flot est maximum et la coupe est de capacité minimale.

Remarques :

-Un flot non complet ne peut pas être maximum

φ maximum $\Rightarrow$ φ complet, mais φ complet $\not\Rightarrow$ φ maximum ,

3.2 Algorithme de Ford-Fulkerson

L'algorithme de Ford-Fulkerson démarre avec **un flot réalisable** à travers le réseau ; ce flot est amélioré itérativement.
Dans le cas où ce flot est maximum, il permet de déterminer la fonction de flot satisfaisant cette valeur ainsi que la coupe minimale.
Dans le cas où le flot n'est pas maximum, il a pour objectif de mettre en évidence une chaine améliorante correspondant à ce flot.
L'algorithme comporte deux phases. La première consiste à rechercher la chaine améliorante qui permet d'augmenter le flot. Si une telle chaine existe, la seconde phase montre comment

modifier la fonction de flot de manière à augmenter le flot. Sinon, la fonction de flot actuelle est optimale et le calcul est terminé.

3.2.1 Marquage utilisé

L'algorithme, qui est basé sur un processus de marquage, utilise pour les sommets des labels λ comme suit : $\lambda(x)=(x, sens, \varepsilon_x)$

- x et y sont des sommets du réseau de transport.
- sens est + ou – ou / s'il est indéfini
- ε est un nombre réel ≥ 0 ou ∞

Lorsqu'un sommet y porte ce label, ceci signifie qu'il existe une chaine *Ch* : S–y qui contient l'arc
(x, y) et dont le flot est $\varepsilon(Ch)$. La direction est + si l'arc est dans le sens du chemin, et – dans le cas contraire.

Un sommet y est marqué si le sommet x a été marqué. Deux cas peuvent se présenter :

- Si a = (x, y) est un arc et $\lambda(x)=(.,.,\varepsilon_x)$ et si , $\varphi(a)<c(a)$, alors le label du sommet y est $\lambda(y)=(x,+,\varepsilon_y)$ où $\varepsilon_y=min(\varepsilon_x, c(a)-\varphi(a))$

- Si a = (y, x) est un arc et $\lambda(x)=(.,.,\varepsilon_x)$ et si $\varphi(a)>0$, alors le label du sommet y est $\lambda(y)=(x,+,\varepsilon_y)$ où $\varepsilon_y=min(\varepsilon_x, \varphi(a))$

Procédure de marquage

Début
Marquer [+] le sommet S
Répéter
- sélectionner un sommet marqué x,
- marquer [+] tout sommet y non marqué, extrémité terminale d'un arc (x, y) non saturé
- marquer [-] tout sommet y non marqué, l'extrémité initiale d'un arc (y, x) tel que b(y, x) > b(y, x)

jusqu' à ce que (aucun sommet ne peut être marqué) ou (P est marqué)

Si P n'est pas marqué alors le flot est maximum
sinon le flot peut être amélioré
fin si

Fin

3.2.2 Principe de l'algorithme

Lors de la première phase de l'algorithme, les sommets qui peuvent être marqués au maximum une fois, sont marqués avec les labels indiqués ci-dessus.
La procédure de marquage se termine lorsque le sommet P est marqué ou bien lorsqu'on ne peut plus marquer de sommet.
Dans le premier cas, le flot obtenu n'est pas maximum et l'algorithme passe à la seconde phase pour augmenter le flot en utilisant les labels des sommets de la chaine Ch obtenue lors de la phase 1. Dans le second cas, le flot obtenu est maximum et on s'arrête. L'ensemble des

sommets marqués déterminent la coupe minimale.

Algorithme : Ford-Fulkerson (G, ,)

Etape 1 : Initialisation

/* Marquer le sommet S*/

$\lambda(S)$ = (/,/, /)

/* Choisir un flot initial φ_0 qui satisfait la relation de conservation/

$\varphi \leftarrow \varphi_0$;

Etape 2 : Calcul du flot complet

Considérer tous les chemins de S vers P et

Augmenter le flot , φ_0 de façon à avoir un flot complet

Etape 3 : Calcul du flot max et recherche de la coupe minimalePhase 1 : Procédure de marquage

Tant que il existe un sommet y non marqué, faire
marquer y selon l'une des deux conditions :
- (x, y) ∈ A, x marqué et $\varphi(x,y)<c(x,y)$ ou
- (y, x) ∈ A, x marqué et $\varphi(x,y)$ > 0
Fin tant que

Phase 2 : Augmenter le flot

Si P est marqué Alors
Améliorer (φ ,*Ch*)
Enlever les labels des sommets et aller à la Phase 1
Sinon
Déterminer (C) /* Définir la coupe minimale */
Max (φ) = Min (c (C))
Fin si

Etape 4 : Resultats

$\varphi \leftarrow \varphi_{max}$ /* φ , est la fonction de flot satisfaisant , φ_{max} */

c (C_{min}) /* C_{min} est la coupe minimale */

Fin

La procédure Améliorer (φ , *Ch*) est la procédure qui permet d'améliorer le flot en utilisant les labels des sommets du chemin *Ch*. (Recherche de la chaine améliorante)

Procédure pour améliorer (φ , *Ch*)

x ← P ;
Tant que x ≠ S faire
Déterminer ε_T
Si $\lambda(x)=(z,+,\varepsilon_x)$ Alors
$\varphi(z,x) \leftarrow \varphi(z,x)+\varepsilon_T$; x ← z ;
Sinon /* $\lambda(x)=(z,-,\varepsilon_x)$ */
$\varphi(x,z) \leftarrow \varphi(x,z)-\varepsilon_T$; x ← z ;
Fin si
Fin tant que
Fin

Exemple :

On donne le réseau de transport de la Figure 2.9, appliquer l'algorithme de Ford-Fulkersonafin de déterminer le flot maximal que l'on peut faire passer à travers ce réseau.

Etape 1 : Initialisation

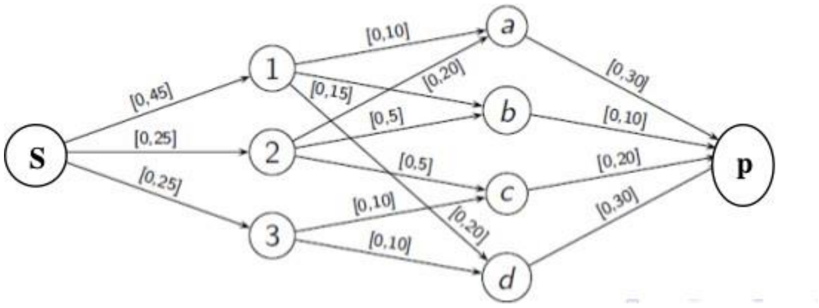
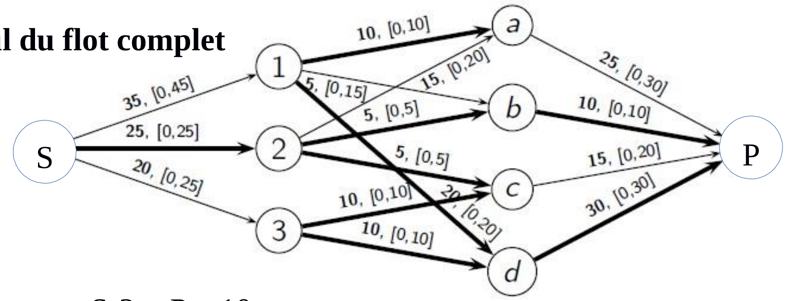


Figure 2.13. Réseau de transport.

Construire un flot complet lors de la première étape de l'algorithme → on part d'un flot complet au lieu de φ_0 (Partir d'un flot réalisable). Examiner tous les chemins de S à P de façon systématique,

Pour chaque chemin *Ch* de S à P, faire passer un flot égal à la capacité résiduelle minimale des arcs de *Ch.*

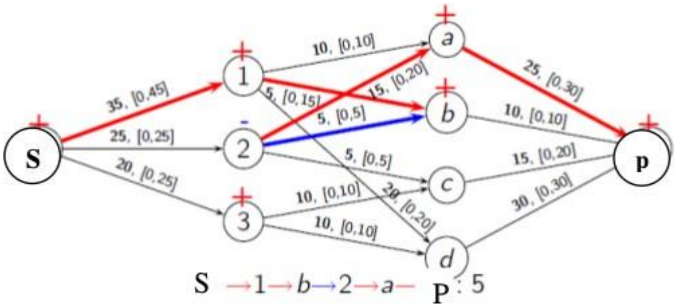
Etape 2 : Calcul du flot complet



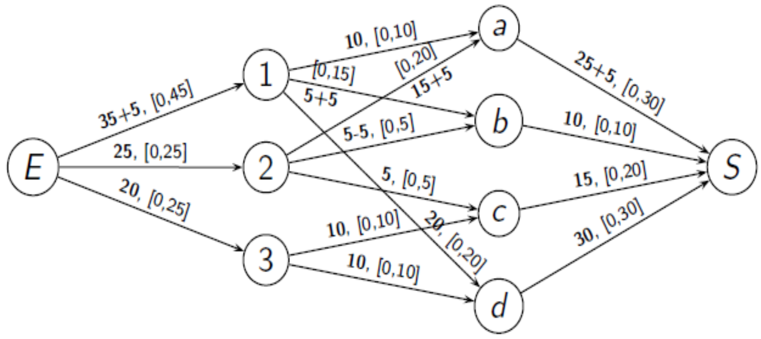
S-3-d-P :10 S-3-c-P :10
S-2-c- P : 5 S-2-b- P : 5
S-2-a- P :15 S-1-d- P : 20 Total = 80
S-1-b- P : 5 S-1-a- P :10

Etape 3:

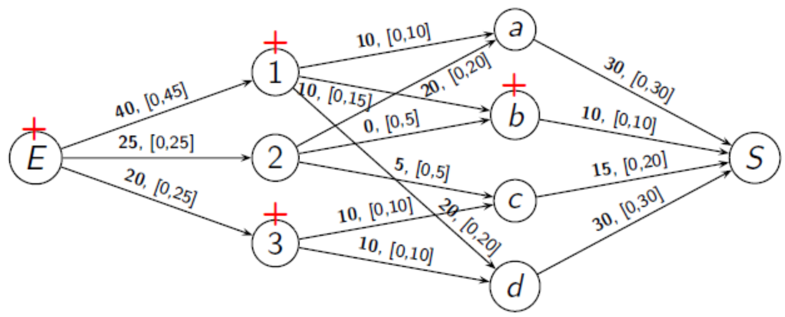
Phase 1 : Procédure de marquage



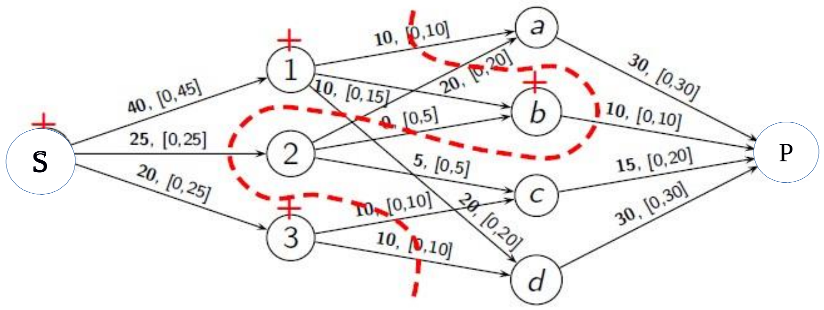
Phase 2 : Augmenter le flot procédure de changement de flot (rechercher une chaîne améliorante) puis reprendre la phase une de marquage.



Procédure de marquage :



Etape 4 : Resultats (Le flot maximum et la coupe minimale)



On ne peut plus marquer de sommets, P n'est pas marqué → le flot est maximum,

$v(\varphi_{max}) = 85.$

La coupe de valeur minimale est engendrée par le sous-ensemble A des sommets marqués (S , 1 , b, 3)

$$C_{min} = 85.$$

Partie 04: Problème de l'arbre de poids minimal

1. Problème

Dans une ville possédant plusieurs quartiers on veut construire un réseau d'alimentation en eau reliant tous les quartiers et qui coute le moins cher possible. Ce problème peut être représenté par un graphe dont les sommets sont les différents quartiers et les arcs sont les branches du réseau qui relient deux quartiers. Chaque branche possède une valeur qui représente son cout de construction. La solution revient à déterminer un graphe partiel qui soit connexe (pour relier tous les quartiers) et qui ne possède pas de cycles (ces derniers sont inutiles et entrainent un cout d'installation supplémentaire). On cherche donc un arbre de poids minimal.

2. Résolution du problème

Ce problème en théorie des graphes correspond à la recherche d'un arbre couvrant de poids minimum. Il se pose pour les graphes dont les arcs sont valués. On appelle ces valeurs des poids. L'ensemble des connexions potentielles peut être représenté par un graphe $G = (V,E)$ dans lequel chaque arête e est associée à un cout $C(e)$ positif. Connecter tout les quartiers correspond à sélectionner un ensemble d'arêtes F de G tel que le graphe partiel $H = (V, F)$ induit est connexe. Le poids de cette solution $C(H)$ est définie comme la somme des poids de ses arêtes.

Le problème s'énonce donc :
Etant donne un graphe $G = (V,E)$, déterminer un graphe partiel connexe $H = (V, F)$ de poids minimum.

Il est facile de voir qu'un tel graphe partiel H doit être un arbre :
Si H n'est pas acyclique, on peut supprimer une arête d'un de ses cycles sans le déconnecter et en faisant diminuer le poids total.

3. Arbre couvrant

Définition 4 Un arbre couvrant pour un graphe G = (V,E) est un arbre construit uniquement à partir des arêtes de E et qui connecte ("couvre") tous les sommets de V .
Un arbre couvrant d'un graphe G est donc un graphe T tel que :
– Le graphe T est un arbre.
– Le graphe T est un graphe partiel de G.

4. Arbre couvrant de poids minimum

Le problème de l'arbre couvrant de poids minimum consiste à trouver un arbre couvrant dont la somme des poids c(e) des arêtes est minimum.
La seule condition, nécessaire et suffisante, pour qu'un graphe admette un arbre couvrant est qu'il soit connexe.
Plusieurs algorithmes ont été proposés pour la résolution du problème de l'arbre couvrant de poids minimum (Minimum Spanning Tree, MST).
Dans ce qui suit, sont présentées deux méthodes: l'algorithme de Kruskal et l'algorithme de Prim .

– ***Algorithme de Kruskal*** Il maintient au fur et à mesure de la construction un graphe partiel acyclique. Il se base sur la caractérisation des arbres, étant des graphes acycliques maximaux.

– ***Algorithme de Prim*** : Il maintient au fur et à mesure de la construction d'un sous-graphe connexe qui grossit petit a petit.

4.1 Algorithme de KRUSKAL

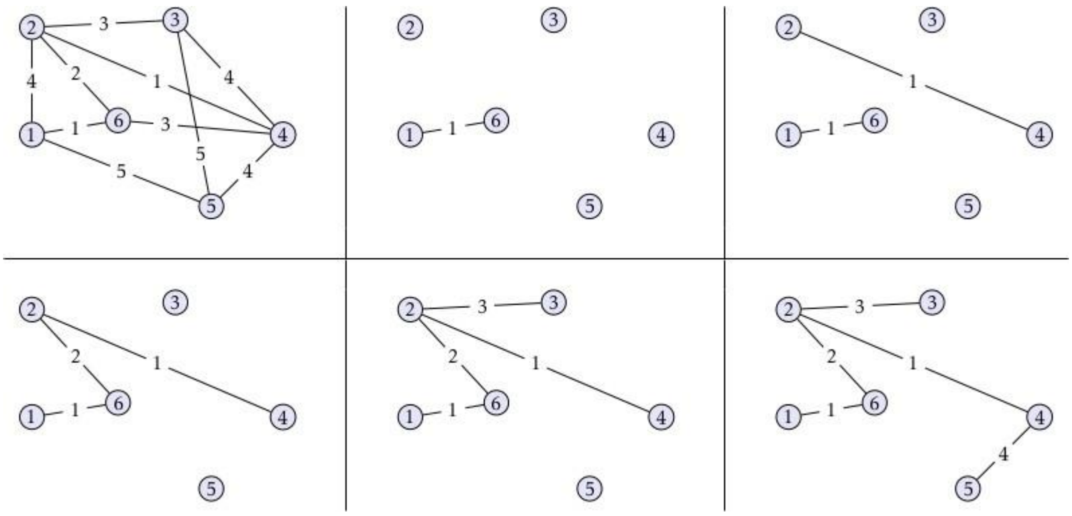
Cet algorithme mis au point par J. Kruskal permet de trouver un arbre de poids minimal pour un graphe connexe et valué. Il retrouve un sous ensemble d'arcs qui forment un arbre touchant tous les sommets. Si le graphe n'est pas connexe, l'application de l'algorithme permet de trouver une forêt couvrante maximale de poids minimum. (Un arbre de poids minimal pour chaque CC).

ALGORITHME Kruskal
ENTREES Graphe ***G=(V,E)***, ***c*** une valuation positive des arêtes
SORTIE ***T*** une forêt couvrante maximale de poids minimum

e : TABLEAU des arêtes du graphe ***G***
F : ENSEMBLE des arêtes de la forêt
Initialiser ***F*** à vide
Trier les arêtes de ***e*** par poids **c(e[i])** croissant.
Pour ***i*** = 1 à |***E*** |
    Si ***F*** $\cup$ **{e[i]}** est acyclique Alors
        F := ***F*** $\cup$ **{e[i]}**
    Fin Si
Fin Pour
Retourner ***T=(V,F)***

Remarques :
- Pour trouver l'arbre de poids maximal, il suffit d'ordonner les arcs dans un ordre décroissant selon leur poids.
- Si toutes les valeurs sont différentes, l'arbre est unique.

Exemple 1:



4.2 Algorithme de PRIM

L'algorithme de Prim se base sur la caractérisation des arbres comme des graphes connexes minimaux: on ne peut enlever une arête à un arbre sans le déconnecter.

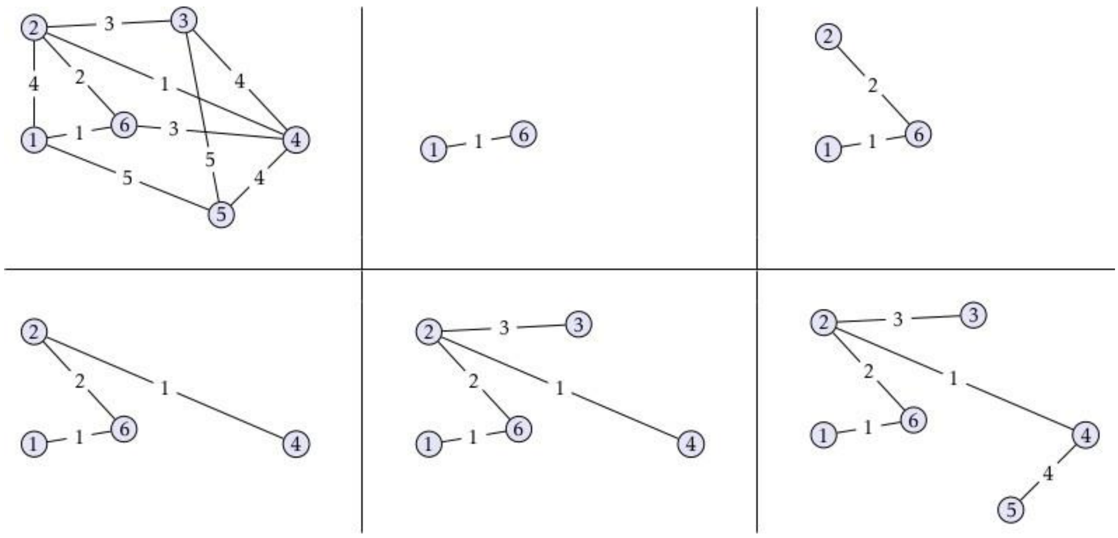
L'idée de l'algorithme est de maintenir un sous-graphe partiel connexe, en connectant un nouveau sommet à chaque étape. L'algorithme de Prim va ainsi faire grossir un arbre jusqu'à ce qu'il couvre tous les sommets du graphe.

ALGORITHME Prim
ENTREES ***G=(V,E)*** un graphe connexe avec une valuation positive des arêtes
SORTIE ***T*** un arbre couvrant de poids minimum

F : ENSEMBLE des arêtes de l'arbre
Initialiser ***F*** à vide
Marquer arbitrairement un sommet

Tant Que il existe un sommet non marqué adjacent à un sommet marqué
    Sélectionner un sommet ***y*** non marqué adjacent à un sommet marqué ***x***
    tel que ***(x,y)*** est l'arête sortante de plus faible poids
    F := ***F*** $\cup$ **{*(x,y)*}**
    Marquer ***y***
Fin TantQue
Retourner ***T=(V,F)***

Exemple 1: Un autre exemple d'application de l'algorithme de Prim à partir du sommet 1.



Chapitre 03
Programmation linéaire

Objectifs:

- La formulation d'un programme linéaire et ses méthodes de résolution.

Dans ce cours Vous allez voir trois parties:

- **Partie 01:** La formulation d'un programme linéaire et sa représentation graphique;
- **Partie 02:** La résolution graphique ;
- **Partie 03:** La résolution algebrique (Simplex).

Préambule

Dans le contexte de la programmation linéaire, le terme programmation désigne l'organisation des calculs et non la réalisation d'un programme informatique. Du point de vue applications, l'optimisation linéaire est d'une grande portée. Elle s'applique à des problèmes très variés qui sont issus de l'économie, de l'ingénierie, de la physique ou encore des modèles probabilistes. Dans ce cadre, on peut citer par exemple, les problèmes de type gestion de stock, gestion de production, transport de marchandise, affectation du personnel, systèmes industriels, réseaux de communication, etc. Pour les modèles de programmation linéaire, on est souvent amené à maximiser un gain ou minimiser un coût. Ceci explique d'ailleurs pourquoi la fonction à maximiser s'appelle fonction d'objectif ou économique

Objectifs

L'objectif de cette partie est de voir comment on peut résoudre des problèmes où on veut maximiser ou minimiser cette fonction dépendant de plusieurs variables qui sont soumises à plusieurs contraintes. Le nom du domaine qui traite ce genre de problème est la recherche opérationnelle.

La recherche opérationnelle (aussi appelée aide à la décision) peut être définie comme l'ensemble des méthodes et techniques rationnelles orientées vers la recherche de la meilleure façon d'opérer des choix en vue d'aboutir au meilleur résultat possible. Quant à l' optimisation, c'est la recherche du maximum (ou du minimum) d'une fonction ainsi que des valeurs des variables qui maximisent (ou minimisent) la fonction.

Partie 01: Formulation d'un programme linéaire (PL)

1. Introduction

L'importance de l'optimisation et la nécessité d'un outil simple pour modéliser des problèmes de décision que soit économique, militaire ou autres ont fait de la programmation linéaire un des champs de recherche les plus actifs au milieu du siècle précédent. Les premiers travaux (1947) sont ceux de George B. Dantzig et ses associés du département des forces aérie des Etats Unis d'Amérique.

Les problèmes de programmations linéaires sont généralement liés à des problèmes d'allocations de ressources limitées, de la meilleure façon possible, afin de maximiser un profit ou de minimiser un coût. Le terme meilleur fait référence à la possibilité d'avoir un ensemble de décisions possibles qui réalisent la même satisfaction ou le même profit. Ces décisions sont en général le résultat d'un problème mathématique.

2. Les conditions de formulation d'un PL

La programmation linéaire comme étant un modèle admet des hypothèses (des conditions) que le décideur doit valider avant de pouvoir les utiliser pour modéliser son problème. Ces hypothèses sont:

- Les variables de décision du problème sont positives
- Le critère de sélection de la meilleure décision est décrit par une fonction linéaire de ces variables, c'est à dire, que la fonction ne peut pas contenir par exemple un produit croisé de deux de ces variables. La fonction qui représente le critère de sélection est dite fonction objectif (ou fonction économique).
- Les restrictions relatives aux variables de décision (exemple: limitations des ressources) peuvent être exprimées par un ensemble d'équations linéaires. Ces équations forment l'ensemble des contraintes.
- Les paramètres du problème en dehors des variables de décisions ont une valeur connue avec certitude.

3. Les étapes de formulation d'un PL:

Généralement il y a trois étapes à suivre pour pouvoir construire le modèle d'un programme linéaire:

- Identifier les variables du problème à valeur non connues (variable de décision) et les représenter sous forme symbolique (exp. x_1, y_1).
- Identifier les restrictions (les contraintes) du problème et les exprimer par un système des inégalités linéaires.
- Identifier l'objectif ou le critère de sélection et le représenter sous une forme linéaire en fonction des variables de décision. Spécifier si le critère de sélection est à maximiser ou à minimiser.

4. Présentation Théorique

Un programme linéaire consiste à trouver le maximum ou le minimum d'une forme linéaire dite fonction objectif en satisfaisant certaines équations et inégalités dites contraintes. En langage mathématique, on décrira de tels modèles de la manière suivante :

Soient N variables de décision x_1, x_2,…, x_n, l'hypothèse que les variables de décision sont positives implique que $x_1 \geq 0, x_2 \geq 0, \ldots, x_N \geq 0$.

La fonction objectif est une forme linéaire en fonction des variables de décision de type

$$z = c_1 x_1 + c_2 x_2 + \ldots + c_N x_N$$

où les coefficients $c_1, \ldots, c_N$ doivent avoir une valeur constante bien déterminée (avec certitude) etpeuvent être positifs, négatifs ou nuls. Par exemple le coefficient c_i peut représenter un profitunitaire lié à la production d'une unité supplémentaire du bien x_i, ainsi la valeur de z est leprofit total lié à la production des différents biens en quantités égales à $x_1, x_2, \ldots, x_N$. Supposons que ces variables de décision doivent vérifier un système d'équations linéairesdéfinis par M inégalités

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 + \ldots + a_{1N} x_N &\geq b_1 \\ a_{21} x_1 + a_{22} x_2 + \ldots + a_{2N} x_N &\geq b_2 \\ &\vdots \\ a_{M1} x_1 + a_{M2} x_2 + \ldots + a_{MN} x_N &\geq b_M \end{aligned}$$

où les coefficients $a_{11}, \ldots, a_{MN}$ et $b_1, \ldots, b_M$ doivent avoir une valeur constante bien déterminée (avec certitude) et peuvent être positifs, négatifs ou nuls. Le paramètre b_j représente la quantité de matière première disponible pour produire x_i unités du produit i, le processus utilise $a_{ij}x_i$ unités de la ressource j.
En suivant les étapes de formulation ci-dessus, on peut représenter le PL comme suit :

$$\begin{aligned} \text{Max} \quad & c_1 x_1 + c_2 x_2 + \ldots + c_N x_N \\ s.c \quad & a_{11} x_1 + a_{12} x_2 + \ldots + a_{1N} x_N \geq b_1 \\ & a_{21} x_1 + a_{22} x_2 + \ldots + a_{2N} x_N \geq b_2 \\ & \vdots \\ & a_{M1} x_1 + a_{M2} x_2 + \ldots + a_{MN} x_N \geq b_N \\ & x_1 \geq 0, x_2 \geq 0, \ldots, x_N \geq 0 \end{aligned}$$

5. Exemples de formulations

Limité au départ aux problèmes industriels et militaires, de nos jours plusieurs problèmes de divers domaines sont représentés ou approximés par des modèles de PL. L'utilisation de ces techniques de modélisation s'est renforcée encore après avoir construit des algorithmes et des logiciels capables de résoudre de plus larges problèmes avec autant de variables de décision que de contraintes.
La tâche de formulation demande généralement une certaine expertise et connaissance du problème pour pouvoir relever facilement les différentes composantes du problème et ainsi donner un programme qui modélise au mieux la situation réelle. Dans ce qui suit, on présentera quelques exemples de formulation en programme linéaire liés à différents problèmes de décision.

Exemple 1: Problème d'agriculture

Un agriculteur veut allouer 150 hectares de surface irrigable entre culture de tomates et celles de piments. Il dispose de 480 heures de main d'œuvre et de 440 m^3 d'eau. Un hectare de tomates demande 1 heure de main d'œuvre, 4 m^3 d'eau et donne un bénéfice net de 100 dinars. Un hectare de piments demande 4 heures de main d'œuvre, 2 m^3 d'eau et donne un bénéfice net de 200 dinars.
Le bureau du périmètre irrigué veut protéger le prix des tomates et ne lui permet pas de cultiver plus de 90 hectares de tomates. Quelle est la meilleure allocation de ses ressources?
Formulation du problème en un PL:

Etape 1: Identification des variables de décision. Les deux activités que l'agriculteur veut déterminer quelles sont les surfaces à allouer pour la culture de tomates et de piments:

- x_1: la surface allouée à la culture des tomates
- x_2: la surface allouée à la culture des piments

On vérifie bien que les variables de décision x_1 et x_2 sont positives: $x_1 \geq 0,\ \ x_2 \geq 0$.

Etape 2: Identification des contraintes. Dans ce problème les contraintes représentent la disponibilité des facteurs de production:

- Terrain: l'agriculteur dispose de 150 hectares de terrain, ainsi la contrainte liée à la limitation de la surface de terrain est $x_1 + x_2 \leq 150$
- Eau: la culture d'un hectare de tomates demande 4 m^3 d'eau et celle d'un hectare de piments demande $2m^3$ mais l'agriculteur ne dispose que de $440m^3$. La contrainte qui exprime les limitations des ressources en eau est $4x_1 + 2x_2 \leq 440$.
- Main d'œuvre: Les 480 heures de main d'œuvre seront départagées (pas nécessairement en totalité) ente la culture des tomates et celles des piments. Sachant qu'un hectare de tomates demande une heure de main d'œuvre et un hectare de piments demande 4
  heures de main d'œuvre alors la contrainte représentant les limitations des ressources humaines est $x_1 + 4x_2 \leq 480$
- Les limitations du bureau du périmètre irrigué: Ces limitations exigent que l'agriculteur ne cultive pas plus de 90 hectares de tomates. La contrainte qui représente cette restriction est $x_1 \leq 90$.

Etape 3: Identification de la fonction objectif. La fonction objectif consiste à maximiser le profit a apporté par la culture de tomates et de piments. Les contributions respectives 100 et 200, des deux variables de décision x1 et x2 sont proportionnelles à leur valeur. La fonction objectif est donc $z = 100x_1 + 200x_2$.

Le programme linéaire qui modélise le problème d'agriculture est:

$$
\begin{array}{ll}
\text{Max} & 100x_1 + 200x_2 \\
s.c. & x_1 + x_2 \leq 150 \quad 4 \\
 & x_1 + 2x_2 \leq 440 \\
 & x_1 + 4x_2 \leq 480 \\
 & x_1 \leq 90 \\
 & x_1 \geq 0,\ \ x_2 \geq 0
\end{array}
$$

Exemple 2: Problème de médecine

Un spécialiste en médecine a fabriqué un médicament (des pilules) pour guérir les sujets atteints d'un rhume. Ces pilules sont fabriquées selon deux formats:
Petite taille: elle contient 2 grains d'aspirine, 5 grains de bicarbonate et 1 grain de codéine.
Grande taille: elle contient 1 grain d'aspirine, 8 grains de bicarbonate et 6 grains de codéine.
Pour guérir la maladie, le sujet a besoin de 12 grains d'aspirine, 74 grains de bicarbonate et 24 grains de codéine. Déterminer le nombre de pilules minimales à prescrire au sujet pour qu'il soit guérit.
Formulation du problème en un PL:

Le problème de médecine présente certaines ressemblances avec le problème de l'agriculture, dans les deux cas c'est un problème d'allocation de ressources.
Etape 01: Les variables de décision qui représentent des valeurs inconnues par le décideur qui est dans ce cas le spécialiste en médecine sont:

- x_1: le nombre de pilules de petite taille à prescrire.

- x_2: le nombre de pilules de grande taille à prescrire.

On vérifie bien que les variables de décision x_1 et x_2 sont positives: $x_1 \geq 0,\ x_2 \geq 0$.

Etape 02: Les contraintes imposées par le problème sur les valeurs possibles de x_1 et x_2 sont:

11) La prescription doit contenir des pilules avec au moins 12 grains d'aspirine. Sachant qu'une petite pilule contient 2 grains d'aspirine et qu'une grande pilule contient un seul grain d'aspirine, on obtient la contrainte suivante : $2x_1 + x_2 \geq 12$.
12) De la même façon que pour l'aspirine, la prescription du spécialiste en médecine doit contenir au moins 74 grains de bicarbonate. Ainsi la contrainte suivante doit être satisfaite: $5x_1 + 8x_2 \geq 74$.
13) Finalement la contrainte imposée par le fait que la prescription doit contenir au moins 24 grains de codéine est $x_1 + 6x_2 \geq 24$.

Etape 3: Identification de la fonction objectif. On remarque qu'il y a plusieurs couples de solutions (x_1, x_2) qui peuvent satisfaire les contraintes spécifiées à l'étape 2. La prescription doit contenir le minimum possible de pilules. Donc le critère de sélection de la quantité de pilules à prescrire est celle qui minimise le nombre total des pilules $z = x_1 + x_2$.

Le programme linéaire qui modélise ce problème médical est donc le suivant:

$$\begin{array}{ll} \text{Min} & x_1 + x_2 \\ s.c. & 2x_1 + x_2 \geq 12 \\ & 5x_1 + 8x_2 \geq 74 \\ & x_1 + 6x_2 \geq 24 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

6. Différentes formes d'un PL

6.1. Forme canonique

Un programme linéaire (PL) est dit sous forme canonique pure s'il s'écrit:

$$\text{Max } Z = \sum_{j=1}^{n} c_j\, x_j$$
$$\sum_{j=1}^{n} a_{ij}\, x_j \leq b_i \qquad b_i \geq 0$$
$$x_j \geq 0 \quad j = 1,\ldots,n$$

Ou encore: $\text{Max } Z = c\,x$
$$A\,x \leq b$$
$$x \geq 0$$

6.2. Forme standard

Un programme linéaire (PL) est dit sous forme standard s'il s'´ecrit:

$$\text{Max } Z = \sum_{j=1}^{n} c_j\, x_j$$
$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i \qquad b_i \geq 0$$
$$x_j \geq 0 \quad j = 1,\ldots,n$$

Ou encore:
$$\text{Max } Z = c\,x$$
$$A\,x = b$$
$$x \geq 0$$

Remarque: Dans la forme standard toutes les contraintes sont sous forme d'égalité.

6.3. Forme mixte

Même fonction objectif que les deux précédentes formes sauf que certaines contraintes sont des « = » d'autres sont des inégalités «<= ou >= »

4. Passage à la forme standard/canonique

Tout programme linéaire PL peut se mettre sous la forme standard ou canonique.
En effet, il suffit de faire les transformations suivantes:
T1) Min Z= - Max (-Z)
T2) Lorsque une contrainte est sous forme d'une inégalité. Deux cas se présentent:
$a_{i1} x_1 + \ldots + a_{in} x_n \leq b_i$. On introduit une variable d'écart e_i et on obtient:
$a_{i1} x_1 + \ldots + a_{in} x_n + e_i = b_i$
$a_{i1} x_1 + \ldots + a_{in} x_n \geq b_i$. On introduit une variable d'éxcés e_i et on obtient: $a_{i1} x_1 + \ldots + a_{in} x_n - e_i = b_i$
T3) Lorsqu'une contrainte est sous forme d'une égalité:

$$a_{i1} x_1 + \ldots + a_{in} x_n = b_i$$

On la transforme en deux inégalités

$$a_{i1} x_1 + \ldots + a_{in} x_n \leq b_i$$

et

$$a_{i1} x_1 + \ldots + a_{in} x_n \geq b_i \iff -a_{i1} x_1 + \ldots - a_{in} x_n \leq -b_i$$

T4) Si un bi est négatif

$$a_{i1} x_1 + \ldots + a_{in} x_n = b_i \quad \text{avec } b_i < 0$$

$$- a_{i1} x_1 \ldots - a_{in} x_n = - b_i \text{ et } - b_i > 0$$

T5) Si une variable x_j est quelconque

On pose $x_j = x^1_j - x^2_j$ avec $x^1_j \geq 0$ et $x^2_j \geq 0$

Exemple

Soit le PL suivant

$$\text{Min } Z = 2 x_1 + x_2 - x_3 + x_4$$
$$3 x_1 - x_2 \geq 5$$
$$x_1 + 3 x_4 \leq 8$$
$$2 x_1 - x_3 \geq 1$$

Ecrire la forme standard et la forme canonique de ce PL

Forme standard:

Min Z <=> (-)Max (-Z) . On introduit les variables d'excés e_1, e_2 et e_3 aux trois contraintesrespectives et on obtient

$$\text{Max } Z' = - 2 x_1 - x_2 + x_3 - x_4$$
$$3 x_1 - x_2 - e_1 = 5$$
$$x_1 + 3 x_4 + e_2 = 8$$
$$2 x_1 - x_3 - e_3 = 1$$

Partie 02:
Résolution de Programmes Linéaires: La méthode graphique

1. Introduction

Après avoir illustré par des exemples, comment un problème pratique peut être modélisé par un programme linéaire, l'étape qui va suivre sera certainement celle de la résolution de ce problème mathématique.

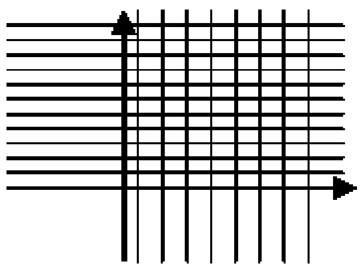
Deux façons de résoudre un problème de programmation linéaire:

<u>Méthode graphique</u>
La méthode graphique est l'une des premières méthodes utilisées. Pour résoudre un problème de programmation linéaire simple ayant seulement deux variables de décisions. Cela permet de visualiser le principe général de la programmation linéaire. Cette méthode est beaucoup trop longue et difficile, voire impossible, à visualiser s'il y a plus de deux variables.

<u>Méthode du simplexe</u>
C'est une méthode algorithmique plus rapide pour résoudre les problèmes comportant plus de 2 variables. Elle est généralement programmée et implémentée à l'ordinateur.
Ceci indique que dans ce chapitre on examinera seulement les programmes linéaires à deux variables de décision.

2. Système d'axes

Une des conditions de la réussite de notre représentation graphique est le choix d'un système d'axes. Un mauvais choix peut rendre notre représentation non claire et imprécise.



A cause des contraintes de non-négativité des variables de décision, nous nous intéressons seulement au cadran positif (voir figure ci-dessus).

Cette région s'appelle la *région des solutions possibles* du problème.

Prenons l'Exemple 2 relatif au problème de médecine. Le programme linéaire est le suivant:

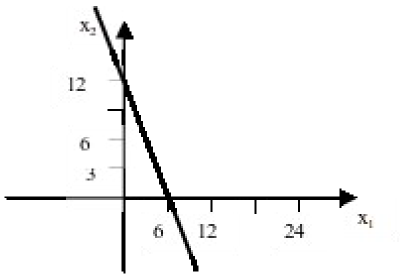
$$
\begin{array}{ll}
\text{Min} & x_1 + x_2 \\
s.c. & 2x_1 + x_2 \geq 12 \\
 & 5x_1 + 8x_2 \geq 74 \\
 & x_1 + 6x_2 \geq 24 \\
 & x_1 \geq 0, x_2 \geq 0
\end{array}
$$

Un bon choix se base sur une lecture des différents paramètres du programme linéaire. Dans notre cas, on ne peut qualifier de bon, le choix de 20 comme unité dans les deux axes.

3. Représentation graphique des contraintes

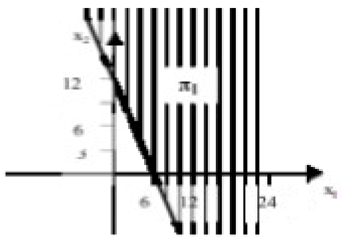
Parmi les solutions possibles d'un problème, il y a celles qui vont satisfaire toutes les contraintes du programme, appelés solutions réalisables, et celles qui vont satisfaire une partie ou aucune de ces contraintes, appelés solutions non réalisables.
Une représentation graphique des inégalités (des contraintes) va nous permettre de déterminer l'ensemble des solutions réalisables.

Revenons à l'Exemple 2 du problème de médecine.
Une des contraintes de ce problème est celle relative au grain d'aspirine:
$2x_1+x_2 \geq 12$.
L'ensemble des solutions qui vérifient cette inégalité est le même que celui qui vérifie $2x_1+x_2=12$ et $2x_1+x_2>12$.

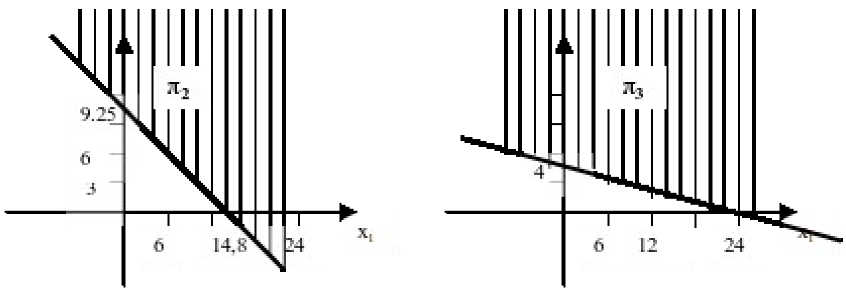


L'ensemble des solutions qui correspond à l'équation est l'ensemble des points de la droite l définie par $x_2=-2\ x_1+12$. Cette droite a une pente égale à –2 et intercepte l'axe des ordonnées en 12 (voir figure ci-dessus).

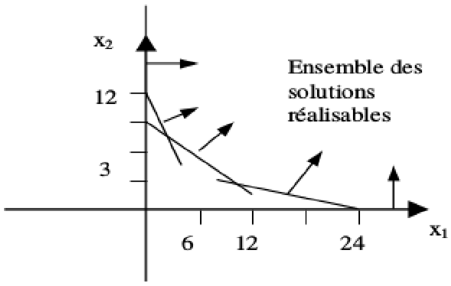
L'inégalité $2\ x_1+x_2>12$ correspond à un demi-plan limité par la droite $x_2=-2\ x_1+12$.
Or cette droite divise le plan en deux demi-plans ouverts donc quel est le demi-plan à choisir?



Pour ce faire, il suffit de prendre un point de l'un des demi-plans (c'est à dire n'appartenant pas à la droite $x_2=-2\ x_1+12$) et voir s'il vérifie l'inégalité $2\ x_1+x_2>12$. Par exemple le point de coordonnées (0,0) ne vérifie pas l'inégalité $2x_1+x_2>12$ donc le demi-plan **Π1** au-dessus de la droite est celui recherché (voir figure ci-dessus).
L'espace hachuré représente le demi-plan fermé des solutions qui vérifient la contrainte $2x_1+x_2>12$.
Si on fait de même pour les deux autres contraintes du problème (voir figures ci-dessous), on obtient les deux autres demi-plans **Π2** et **Π3** relatifs aux solutions vérifiant respectivement les contraintes $5x_1+8x_2 \geq 74$ et $x_1+6x_2 \geq 24$.



Une solution possible du problème est dite réalisable si et seulement si elle vérifie toutes les contraintes, c'est à dire si elle appartient aux trois demi-plans relatifs à chaque contrainte du programme linéaire, en d'autre terme à **Π1** ∩ **Π2 Π3** (voir la figure ci-dessous).



Définition: Un ensemble E non vide est dit convexe si et seulement si pour tout élément *x* et *y* de E et pour tout $\lambda \in [0,1], \lambda x + (1-\lambda) y \in E.$

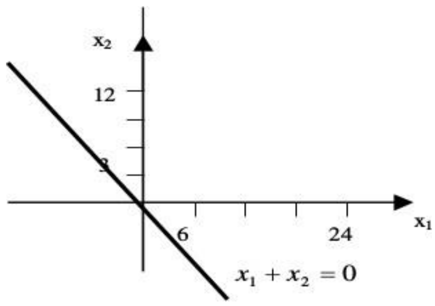
Un objet géométrique est dit convexe lorsque, chaque fois qu'on y prend deux points A et B, le segment [A, B] qui les joint y est entièrement contenu. Ainsi un cube plein, un disque ou une boule sont convexes, mais un objet creux ou bosselé ne l'est pas.

On peut vérifier facilement que chacun des demi-plans Π_1, Π_2 , Π_3 est convexe en vérifiant que pour toute paire de points P_1 et P_2, l'ensemble des points qui forment le segment $[P_1P_2]$ appartient au demi-plan.

En gros, l'intersection d'ensembles convexes (non vide) est convexe.L'ensemble des solutions réalisables (non vide) est convexe.

4. Représentation de la fonction objectif

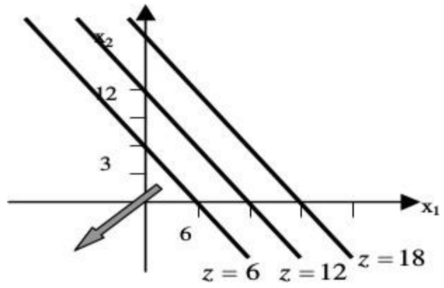
Soit *z* la valeur de la fonction objectif du problème de médecine $z=x_1+x_2$.
Pour z=0, la fonction objectif est représentée de la manière suivante:



Pour z=6, c'est à dire que le nombre de pilules à prescrire est égale à 6 pilules. La fonction objectif est représentée comme suit:

Chaque point du segment qui relie les points (6,0) à (0,6) représente des solutions qui

engendrent une prescription avec 6 pilules des deux tailles.
On peut tracer une infinité de droites qui représentent les différentes valeurs de la fonction objectif, toutes ces droites ont le même coefficient directeur (-1). Par suite elles sont parallèles entre elles. De plus on peut diminuer la valeur de z indéfiniment dans le sens indiqué dans la figure ci-dessous.



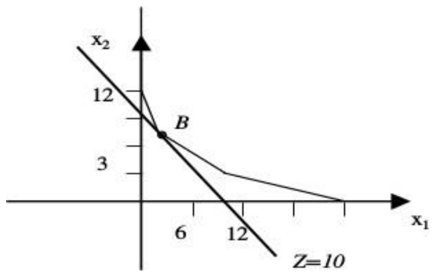
Le problème est de connaître qu'elle est la droite qui correspond à la valeur minimal de la fonction objectif?

5. Recherche de la solution optimale

a. Résolution graphique

Si nous retraçons l'ensemble des droites parallèles relatives à différentes valeurs de la fonction objectif sur la figure qui représente l'ensemble des solutions réalisables, on peut localiser la solution optimale. Elle correspond à la solution réalisable qui intercepte la droite à la plus petite valeur de z.

Dans notre exemple, la solution optimale est l'intersection des deux droites $2x_1 + x_2 = 12$ et $5x_1 + 8x_2 = 74$. La détermination des coordonnées de ce point revient à résoudre le système linéaire suivant:



$$\begin{cases} 2x_1 + x_2 = 12 \\ 5x_1 + 8x_2 = 74 \end{cases}$$

Elle correspond d'après le graphe au point (2,8). Donc la prescription optimale est de 2 pilules de petite taille et 8 pilules de grande taille. Le nombre de pilules (la valeur de la fonction objectif) est égale à 10.

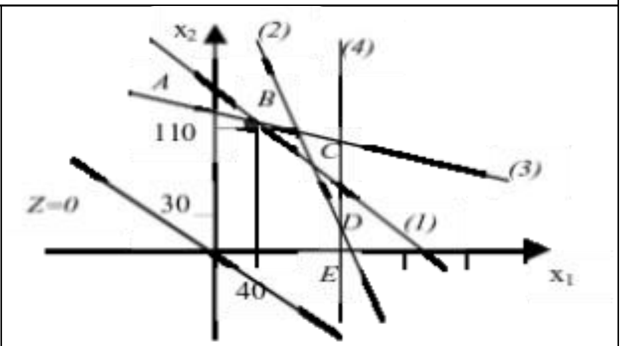
b. Résolution par énumération:

On remarque que la solution optimale du problème de médecine est un point extrême qui se trouve sur le bord de l'ensemble des solutions réalisables. Une telle solution est dite solution réalisablede base.
On peut admettre le résultat suivant: « Si un programme linéaire admet une solution optimale alors il existe une solution réalisable de base à laquelle la fonction objectif atteint la valeur optimale»
Une méthode de résolution du programme linéaire consiste donc à déterminer les solutions réalisables de base (les points d'intersection des droites qui forment les contraintes) et à calculer pour chaque point la valeur de la fonction objectif. La solution du programme linéaire est la solution qui donne la valeur optimale de la fonction objectif.

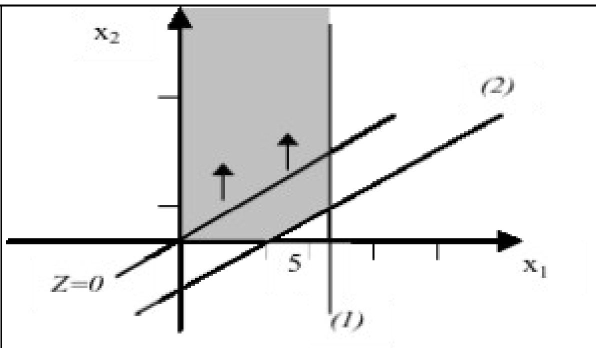
Dans le problème de médecine, l'ensemble des solutions réalisables de base présente 4 points extrêmes A(0,12), B(2,8), C(23/11,126/11) et D(24,0). Les valeurs de la fonction objectif associée respectivement à A, B, C et D sont 12, 10, 149/11 et 24. On vérifie bien que B est la solution optimale du problème avec une valeur optimale égale à 10.

6. Exemples

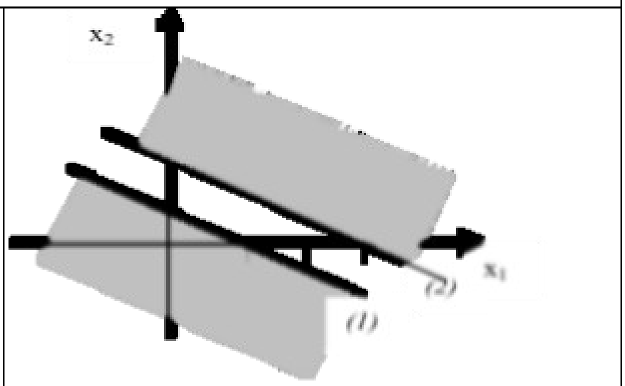
Dans cette section on donne quelques exemples de résolution graphique de problèmes linéaires relatifs au différents cas possibles :

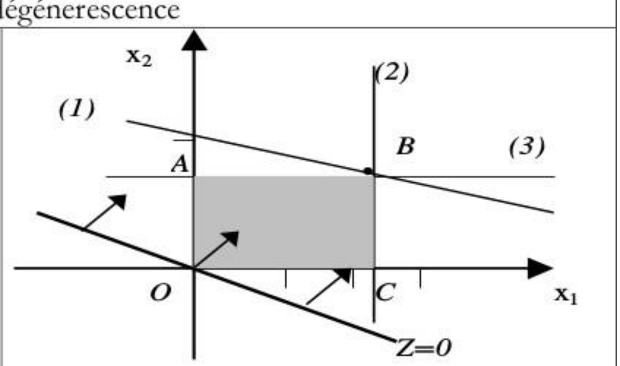
<table>
<tr><th colspan="2">Problème de maximisation</th></tr>
<tr><td>
$$\begin{array}{lll} \text{Max} & 100\,x_1+200\,x_2 & \\ s.c. & x_1+x_2 \leq 150 & (1) \\ & 4\,x_1+2\,x_2 \leq 440 & (2) \\ & x_1+4\,x_2 \leq 480 & (3) \\ & x_1 \leq 90 & (4) \\ & x_1 \geq 0, x_2 \geq 0 & \end{array}$$
</td><td>


</td></tr>
<tr><td colspan="2">la solution optimale est B(40,110)</td></tr>
</table>

<table>
<tr><th colspan="2">Problème avec solution non bornée</th></tr>
<tr><td>
$$\begin{array}{lll} \text{Max} & -2\,x_1+3\,x_2 & \\ s.c. & x_1 \leq 5 & (1) \\ & 2\,x_1-3\,x_2 \leq 6 & (2) \\ & x_1 \geq 0, x_2 \geq 0 & \end{array}$$
</td><td>


</td></tr>
<tr><td colspan="2">On peut augmenter la valeur de la fonction objectif dans la direction des flèches indéfiniment donc la solution est non bornée</td></tr>
</table>

<table>
<tr><th colspan="2">Problème impossible</th></tr>
<tr><td>Min $3x_1+2x_2$
$s.c.$ $x_1+2x_2 \le 2$ (1)
$2x_1+4x_2 \ge 8$ (2)
$x_1 \ge 0, x_2 \ge 0$</td><td>
</td></tr>
<tr><td colspan="2">L'espace des solutions réalisables est vide, il est l'intersection des deux zones grises de la figure ci-dessus</td></tr>
</table>

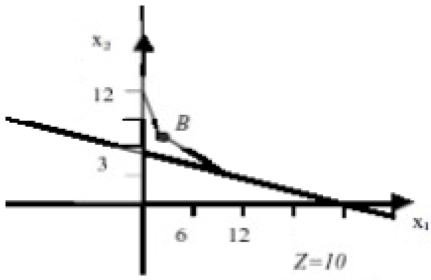
<table>
<tr><th colspan="2">Problème de dégénerescence</th></tr>
<tr><td>Max $x_1 + x_2$
$s.c.$ $3x_1 + 2x_2 \le 40$ (1)
$x_1 \le 10$ (2)
$x_2 \le 5$ (3)
$x_1 \ge 0, x_2 \ge 0$</td><td>
</td></tr>
<tr><td colspan="2">La solution optimale B(10,5) est dite dégénérée si trois contraintes concourent en ce point.</td></tr>
</table>

7. Analyse de sensibilité

Une analyse de sensibilité se résume à la recherche des intervalles de variations possibles des paramètres du programme linéaire sans que la solution optimale ne soit modifiée.

Question: De combien peut-on faire varier la quantité de codéine dans le problème de médecine sans changer la solution optimale.

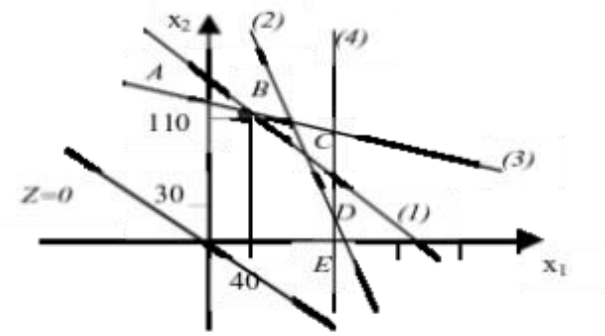
Réponse:



On peut changer la valeur du second membre de la troisième contrainte jusqu'à ce que la droite de coefficient directeur –1/6 touche le point optimal (2,8). C'est à dire qu'on peut varier le second membre de la troisième contrainte de 24 jusqu'à 50 (la valeur 50 est obtenu aprés remplacement de x_1 et x_2 avec les valeurs 2 et 8 respectivement) sans changer la solution optimale.

Question: De combien peut-on faire varier le profit engendré par la culture d'un hectare supplémentaire de tomates, dans le problème de l'agriculture, sans changer la solution optimale?

Réponse:



Soit la variation du profit engendré par la culture d'un hectare supplémentaire de tomate. La fonction objectif est égale à (100 +λ) x1+200 x2

La solution demeure optimale si la pente de la fonction objectif reste toujours comprise entre la pente de la contrainte (1) et (3). Ceci est équivalent à dire que :

$$-1 \leq -\frac{200}{100+\lambda} \leq -\frac{1}{4} \quad - \quad -50 \leq \lambda \leq 100$$

On peut vérifier aussi que si:

- $\lambda < -50$ alors la solution optimale est A
- $\lambda = -50$ alors le problème est à solutions multiples: le segment [AB]
- $-50 < \lambda < 100$ alors la solution optimale est B
- $\lambda = 100$ alors le problème est à solutions multiples: le segment [BC]
- $100 < \lambda < 300$ alors la solution optimale est C
- $\lambda = 300$ alors le problème est à solutions multiples: le segment [CD]
- $\lambda > 300$ alors la solution optimale est D

Partie 03:
Résolution Algébrique: La méthode du Simplexe

Introduction

On a présenté dans le chapitre précédent une procédure graphique pour résoudre un programme linéaire à deux variables. Par contre, dans la plupart des problèmes réels, on a plus que deux variables à déterminer. Une procédure algébrique pour résoudre les programmes linéaires avec plus que deux variables fera l'objet de ce chapitre. C'est la méthode de simplexe. Une implémentation de cette procédure à permis de résoudre des programmes avec un peu plus de quelques milliers de variables.

1. Mise sous forme standard

La mise sous forme standard consiste à introduire des variables supplémentaires (une pour chaque contrainte) de manière a réécrire les inégalités sous la forme d'égalités. Chacune de ces variables représente la quantité de ressources non utilisés. On les appelle variables d'écart.

a) Forme canonique d'un Programme Linéaire

$$
\begin{array}{ll}
\text{Max } z = & c_1 x_1 + c_2 x_2 + \ldots\ldots\ldots + c_n x_n \\
& a_{11} x_1 + a_{12} x_2 + \ldots\ldots\ldots + a_{1n} x_n \leq b_1 \\
& a_{21} x_1 + a_{22} x_2 + \ldots\ldots\ldots + a_{2n} x_n \leq b_2 \\
& \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
& a_{m1} x_1 + a_{m2} x_2 + \ldots\ldots\ldots + a_{mn} x_n \leq b_m \\
& x_1 >= 0 ;\ x_2 >= 0 ;\ \ldots\ldots;\ x_n >= 0
\end{array}
$$

b) Forme standard d'un Programme Linéaire

On transforme les inégalités des contraintes économiques en égalités par introduction de variables supplémentaires positives ou nulles appelées variables d'écart.
$a_{i1} x_1 + a_{i2} x_2 + \ldots\ldots\ldots + a_{in} x_n \leq b_i$ devient
$a_{i1} x_1 + a_{i2} x_2 + \quad + a_{in} x_n + e_i = b_i$

d'où la forme standard

$$
\begin{array}{ll}
\text{Max } z = & c_1 x_1 + c_2 x_2 + \ldots\ldots\ldots + c_n x_n \\
& a_{11} x_1 + a_{12} x_2 + \ldots\ldots\ldots + a_{1n} x_n + e_1 = b_1 \\
& a_{21} x_1 + a_{22} x_2 + \ldots\ldots\ldots + a_{2n} x_n + e_2 = b_2 \\
& \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
& a_{m1} x_1 + a_{m2} x_2 + \ldots\ldots\ldots + a_{mn} x_n + e_m = b_m \\
& x_1 >= 0 ;\ x_2 >= 0 ;\ldots;\ x_n >= 0 ;\ e_1 >= 0 ;\ e_2 >= 0 ;\ldots;\ e_m >= 0
\end{array}
$$

2. Résolution

Afin de comparer avec la résolution graphique, nous pouvons considérer que nous sommes dans un espace à n dimensions (nombre de variables d'activité). Les contraintes délimitent un **polyèdre** convexe, région des solutions admissibles; la fonction objectif est un **hyperplan** que l'on va déplacer le plus loin possible de l'origine, jusqu'à l'extrême limite

où il n'y aura plus qu'un point d'intersection (éventuellement un segment, un plan...) avec la région des solutions admissibles.
La solution se trouvant forcément sur le pourtour du polyèdre admissible, la méthode du simplexe consiste en itérations qui font passer d'un sommet du polyèdre à un autre en sélectionnant le sommet améliorant la fonction objectif. Les étapes du simplexe sont définies par l'organigramme suivant :

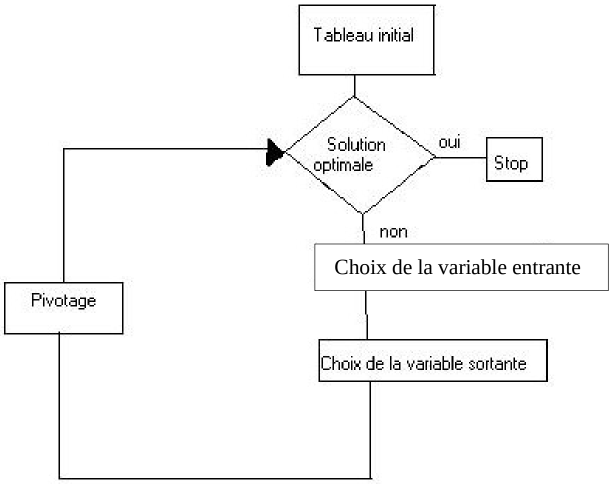


Fig 3.1 Organigramme du simplxe

Pour démarrer l'algorithme, il est nécessaire d'avoir une solution initiale.

Dans ce cas simple, l'origine est solution, c.à.d. que la première solution est $x_1 = 0$; $x_2 = 0$; ....; $x_n = 0$; $e_1 = b_1$; $e_2 = b_2$; ....; $e_m = b_m$ (ceci suppose que les bi ne soient pas négatifs pour satisfaire les contraintes de signe)

L'algorithme, basé sur la méthode du pivot de Gauss pour la résolution des systèmes d'équations linéaires, est présenté sous forme de tableau.

Soit à résoudre le programme linéaire suivant sous sa forme canonique

$$3 x_1 + 4 x_2 \leq 160$$
$$6 x_1 + 3 x_2 \leq 180$$
$$\text{Max } z = 1200 x_1 + 1000 x_2$$
$$x_1 >= 0 \ ; \ x_2 >= 0$$

* Forme standard

$$3 x_1 + 4 x_2 + 1 e_1 + 0 e_2 = 160$$
$$6 x_1 + 3 x_2 + 0 e_1 + 1 e_2 = 180$$
$$\text{Max } z = 1200 x_1 + 1000 x_2 + 0 e_1 \quad 0 e_2$$
$$x_1 >= 0 \ ; \ x_2 >= 0$$

* Tableau 0

En ne conservant que les coefficients des équations ci-dessus, on obtient le tableau de départ

HB B	x_1	x_2	e_1	e_2	C
e1	3	4	1	0	160
e2	6	3	0	1	180
Z	1200	1000	0	0	0

Ce tableau nous donne la première solution admissible:

- Les variables Hors Base, (HB) sont nulles: $x_1 = 0$; $x_2 = 0$ (e_1 et e_2 en rouge ne sont pas hors base; elles ne sont présentes que pour rappeler qu'il s'agit des colonnes des coefficients de ces deux variables)

- Les valeurs des variables dans la Base (B) se lisent dans la colonne C: $e_1 = 160$ et $e_2 = 180$
- La dernière cellule (intersection de C et Z) donne la valeur de -z : -z = 0 donc z = 0
- La ligne Z donne les valeurs marginales ou taux marginal de substitution; elles s'interprètent de la manière suivante: à ce stade de la solution, une augmentation de 1 unité de x1 ferait accroître la fonction objectif de 1200, et une augmentation de 1 unité de x2 ferait accroître la fonction objectif de 1000.

* Tableau 1

On augmente la fonction objectif en faisant entrer une variable dans la base, prenant la place d'une variable qui va sortir de la base.

<u>Critère de sélection de la variable entrant dans la base:</u>

On sélectionne la variable HB ayant le plus grand coefficient positif dans la ligne Z .

x1 entre donc dans la base

Pour sélectionner la variable sortant de la base, il est nécessaire de rajouter une colonne K au tableau, obtenue en faisant le rapport membre à membre de la colonne C et de la colonne de la variable entrant dans la base (x1)

Remarques sur la colonne K:

- Un 0 dans la colonne C est remplacé par un infiniment petit positif e pour effectuer le calcul de K

- Dans la colonne K on ne tient pas compte des valeurs négatives ou indéterminées

HB B	x_1	x_2	e_1	e_2	C	K

e_1	3	4	1	0	160	160/3
e_2	6	3	0	1	180	30
Z	1200	1000	0	0	0	

Critère de sélection de la variable sortant de la base:

On sélectionne la variable dans la Base ayant le plus petit coefficient positif dans la colonne K .

e2 sort donc de la base

HB B	x_1	x_2	e_1	e_2	C	K
e_1	3	4	1	0	160	160/3
e_2	6	3	0	1	180	30
Z	1200	1000	0	0	0	

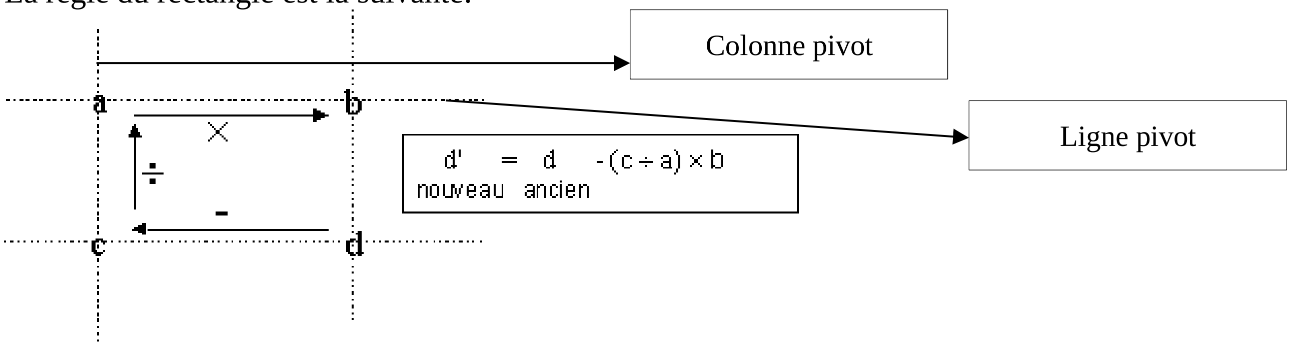
variable sortante

variable entrant

On appelle **pivot** (égal à 6) l'intersection de la colonne de la variable entrante et de la colonne de la variable sortante. Pour obtenir le tableau 1, on applique les règles suivantes:

-Le pivot est égal à 1
- Les coefficients de la ligne du pivot sont divisés par le pivot
- Les coefficients non pivot de la colonne du pivot sont nuls
- Les autres coefficients sont obtenus par la règle du rectangle

La règle du rectangle est la suivante:



<u>Remarque importante:</u>
Si dans la colonne (resp. ligne) du pivot il y a un 0, toute la ligne (resp.colonne) correspondante reste inchangée.

En appliquant ces règles on obtient le tableau 1:

HB B	x_1	x_2	e_1	e_2	C

e_1	0	5/2	1	-1/2	70
x_1	1	1/2	0	1/6	30
Z	0	400	0	-200	-36000

Ce tableau nous donne la deuxième solution admissible:

- Les variables Hors Base (HB), sont nulles: x2 = 0 ; e2 = 0 (x1 et e1 en rouge ne sont pas hors base; elles ne sont présentes que pour rappeler qu'il s'agit des colonnes des coefficients de ces deux variables)
- Les valeurs des variables dans la Base (B) se lisent dans la colonne C: e1 = 70 et x1 =30.
- La dernière cellule (intersection de C et Z) donne la valeur de -z : -z = -36000 donc z =36000 .
- La ligne Z donne les valeurs marginales ou taux marginal de substitution. Elles s'interprètent de la manière suivante: à ce stade de la solution, une augmentation de 1 unité de x_2 ferait accroître la fonction objectif de 400, et une augmentation de 1 unité de e_2 ferait diminuer la fonction objectif de 200 (il est à noter qu'une augmentation de 1 unité de la variable d'écart e_2 revient à diminuer le second membre de l'équation correspondante de 1 unité).

* Tableau 2:

HB B	x_1	x_2	e_1	e_2	C	K	
e1	0	5/2	1	-1/2	70	28	variable sortante
x1	1	1/2	0	1/6	30	60	
Z	0	400	0	-200	- 36000		

variable entrant

d'où le tableau 2

HB B	x_1	x_2	e_1	e_2	C
x2	0	1	2/5	-1/5	28
x1	1	0	-1/5	4/15	16
D	0	0	-160	-120	- 47200

Ce tableau nous donne la troisième solution admissible:

- Les variables Hors Base (HB) sont nulles: $e_1 = 0$; $e_2 = 0$ (x_1 et x_2 en rouge ne sont pas hors base; elles ne sont présentes que pour rappeler qu'il s'agit des colonnes des coefficients de ces deux variables)

- Les valeurs des variables dans la Base (B) se lisent dans la colonne C: x_2= 28 et x_1=16.

- La dernière cellule (intersection de C et Z) donne la valeur de -z : -z = - 47200 donc z =47200 .

- La ligne Z donne les valeurs marginales ou taux marginal de substitution. Elles s'interprètent de la manière suivante: à ce stade de la solution, une augmentation de 1 unité de e1 ferait diminuer la fonction objectif de 160, et une augmentation de 1 unité de e2 ferait diminuer la fonction objectif de 120 (il est à noter qu'une augmentation de 1 unité d'une variable d'écart revient à diminuer le second membre de l'équation correspondante de 1 unité).

<u>Critère d'arrêt des itérations:</u>

Si tous les coefficients de la ligne** Z**, relatifs aux variables HB, sont négatifs ou nuls, la solution trouvée est optimale.

Nous avons donc ici atteint la solution optimale.

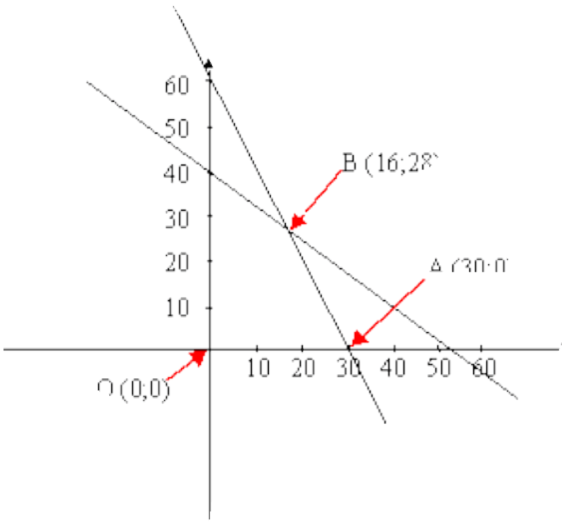
Remarques importantes:

- S'il existe une variable HB ayant un coefficient positif dans la ligne Z et telle que tous les coefficients correspondants dans le tableau soient nuls ou négatifs, alors la solution est infinie.

- Si, à la fin des itérations, une variable est HB avec un coefficient nul dans la ligne Z, alors on a une arête (plan,...) optimale. Les autres sommets solutions sont obtenus en faisant rentrer cette variable dans la base.

Interprétation graphique de la méthode du simplexe:

Les différentes solutions obtenues à chaque tableau correspondent respectivement aux sommets O ($\mathbf{x}_1 = 0$; $\mathbf{x}_2 = 0$), A ($\mathbf{x}_1 = 30$; $\mathbf{x}_2 = 0$), B ($\mathbf{x}_1 = 16$; $\mathbf{x}_2 = 28$) du graphe. On a cheminé sur le pourtour du polyèdre des solutions admissibles, en sélectionnant parmi tous les sommets possibles celui donnant la valeur maximale à la fonction objectif.



3. La dualité

A tout programme linéaire appelé PRIMAL correspond un programme linéaire appelé DUAL obtenu de la manière suivante:

3.1. Définition

La forme d'un programme linéaire de type maximisation est

$$\begin{array}{lll} \text{Max} & z=c^t x & \\ \textit{Contraites} & \text{Ax} \leq b & (\text{PL}1) \\ & x \geq 0 & \end{array}$$

avec x, b , c des vecteurs de dimensions respectives n, m et n , et A une matrice de dimension (m, n)

On appelle programme dual de $(\text{PL}1)$, le programme linéaire suivant :

$$\begin{array}{ll} \text{Min} & w=b^t\ y \\ \textit{Contraintes} & A^t . Y \geq c \\ & y \geq 0 \end{array}$$

avec y un vecteur de dimension m et A^t la transposée de la matrice A .

Le programme $(\text{PL}1)$ est appelé programme Primal.

Pour passer du primal au dual, on remarque que :

- Les termes du second membre des contraintes deviennent les coefficients de la fonction objectif etréciproquement.
- Le problème de maximisation devient un problème de minimisation.
- Les inégalités <= deviennent des inégalités >=
- La matrice A se transforme en sa transposée.

Exemple : Le programme primal (problème de l'agriculteur) est

$$\begin{array}{ll} \text{Max} & z=100\,x_1+200\,x_2 \\ S.C & x_1+x_2 \leq 150 \\ & 4\,x_1+2\,x_2 \leq 440 \\ & x_1+4\,x_2 \leq 480 \\ & x_1 \leq 90 \\ & x_1 \geq 0 \quad , \quad x_2 \geq 0 \end{array}$$

Donc le programme dual est

$$\begin{array}{ll} \text{Min} & w=150\ y_1+440\ y_2+480\ y_3+90\ y_4 \\ S.C & y_1+4\,y_2+y_3+y_4 \geq 100 \\ & y_1+2\,y_2+4\,y_3 \geq 200 \\ & y_1 \geq 0 \quad , \quad y_2 \geq 0 \end{array}$$

PRIMAL **DUAL**

m contraintes d'infériorité	n contraintes de supériorité
n variables d'activité	n variables d'écart
m variables d'écart	m variables d'activité
écriture en ligne	écriture en colonne

Exemple

PRIMAL				DUAL			
3	x1 +	4	x2 ≤ 160	3	y1 +	6	y2 >= 1200
6	x1 +	3	x2 ≤ 180	4	y1 +	3	y2 >= 1000
Max z = 1200	x1 +	1000	x2	Min w = 160	y1 +	80	y2
x1 >= 0 ; x2 >= 0				y1 >= 0 ; y2 >= 0			

A l'optimum, le primal et le dual sont liés par les règles suivantes:

- les fonctions objectifs z et w ont la même valeur optimale
- la valeur marginale d'une variable dans un programme est égale à l'opposé de la valeur optimale de la variable associée dans l'autre programme et réciproquement.

Exemple

PRIMAL	z = 47200	x1		e1	e2
	valeurs optimales	16		0	0
	valeurs marginales	0	0	-160	-120
DUAL	w = 47200	u1	u2	y1	y2
	valeurs optimales	0	0	160	120
	valeurs marginales	-16	-28	0	0

Annexe : (Notions géométriques)

- Un plan est un sous-espace de dimension 2 dans l'espace à 3 dimensions
  Un hyperplan est un sous-espace à n-1 dimensions dans un espace à n dimensions.
- Le carré, le rectangle, le losange ou toute autre figure géométrique dont le pourtour est délimité par des droites est un polyèdre du plan à 2 dimensions
  Le cube, le parrallelipipède ou toute autre figure géométrique dont le pourtour est délimité par des plans est un polyèdre de l'espace à 3 dimensions.
  De manière générale, un polyèdre est une figure géométrique dont le pourtour est délimité par des hyperplans de dimension n-1 dans un espace à n dimensions.

Partie 02:
Travaux dirigées / Corrigées

Série de travaux dirigés N° 1

Exercice 01

On définit une relation R sur l'ensemble des premiers neufs entiers naturels non nuls comme suit:

x R y <=> x est un diviseur de y

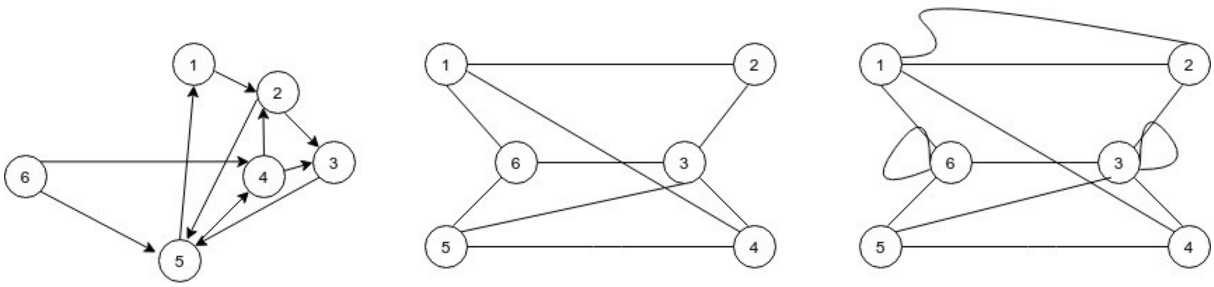
1) Représenter cette relation par un graphe orienté.
2) Déterminer à partir du graphe l'ensemble des nombres pairs et l'ensemble des nombres premiers.

Exercice 02

Soient les graphes a et b en figure 1. Soient les graphes G1, G2 et G3 de la figure 1.

1) Établir la représentation de ces graphes par:
   - (a) Matrice d'adjacence.
   - (b) Matrice d'incidence.
   - (c) Listes des voisins.
   - (d) Les degrés (Intérieurs et Extérieurs pour un graphe orienté).

2) Pour ces différentes représentations, quelle est l'influence de l'ajout (la suppression):
   - (a) D'une arête,
   - (b) D'un sommet et ses arêtes?



(G1) (G2) (G3)

Figure 1. G1: Graphes simple orienté, G2: graphe simple non orienté , G3un multi-graphe non orienté.

Exercice 03

Deux joueurs disposent de deux tas, chaque tas contient trois allumettes, à tour de rôle chaque joueur peut enlever une ou deux allumettes dans un des tas (selon la règle choisie).. Le joueur qui retire la dernière allumette perd la partie.

1- Modéliser le jeu à l'aide d'un graphe.

2- Que doit jouer le premier joueur pour gagner à coup sûr ?

Exercice 04

Modélisez les situations suivantes sous forme de graphes. Dessinez chaque graphe et donnez les matrices de contiguïté correspondantes.

Adnane et Brahim sont amis. Adnane est également amie avec Celia et Dalila. Brahim, Celia et Ékram sont tous amis l'un de l'autre.

Wikipedia a cinq articles particulièrement intéressants: Animal, Terrier, Chili, Désert et Éléphant. Certains d'entre eux sont même liés les uns aux autres!
Il est bien connu qu'aux Pays-Bas, il existe une autoroute à 2 voies d'Amsterdam à Breda, une autre autoroute à 2 voies d'Amsterdam à Cappele aan den IJssel, une autoroute à 3 voies de Breda à Dordrecht, une route à 1 voie de Breda à Ede et une autre de Dordrecht à Ede, et une autoroute à 5 voies de Cappele aan den IJssel à Ede.

Exercice 05

Avez-vous jamais remarqué que dans un groupe de personnes, il y a toujours deux individus qui connaissent exactement le même nombre de membres du groupe?

1. Formaliser la propriété à démontrer dans le vocabulaire des graphes.
2. Démontrer cette propriété (on pourra raisonner par l'absurde et supposer que la propriété à prouver n'est pas vraie pour un graphe à *n* sommets)

Exercice 06

1) Qu'est ce qu'un graphe complet ?
2) Quel est le nombre d'arêtes d'un graphe complet, non orienté G (X,U) avec |X|=n et |U|= m?
3) En déduire qu'un graphe non orienté, simple et ayant plus de (n-1)(n-2)/2 arêtes est connexe
(n nombre de sommets)

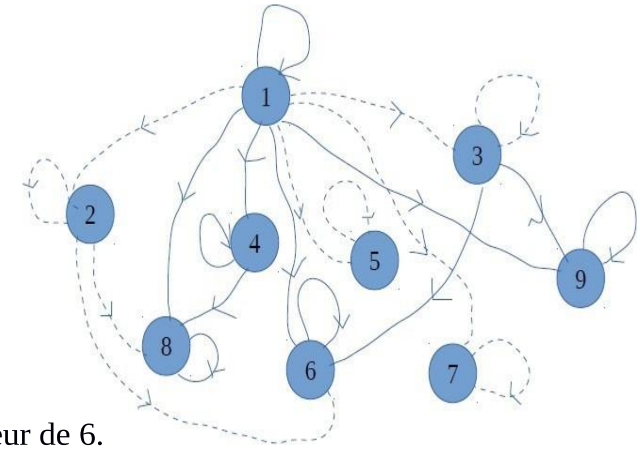
Exercice 07

Répondez pour chacun de ces graphiques: est-il planaire? Est-ce bipartite?

Corrigé des travaux dirigés de la série N° 1

Exercice 01

- Soit X= {1, 2 ,3 , 4, 5, 6, 7, 8, 9} l'ensemble des 9 premiers entiers naturels non nuls. Le graphe représentation la relation représentation la relation R sur l'ensemble X est le graphe G=(X, U) suivant:



On dit qu'il existe un arc reliant un sommet x à un sommet y, si x est diviseur de y, d'où U= {(x,y)/x est un diviseur de y}

Exemple:

L'arc u = (2,6) signifie que 2 est un diviseur de 6.

- Tout nombre **pair** admet le 2 comme diviseur; donc, pour déterminer les nombres pairs du graphe G, il suffit de déterminer tous les successeurs du sommet 2;

C'est à dire: $\Gamma^+ (2) = \{2, 4, 6, 8\}$

Sur le graphe, les nombres pairs sont les extrémités terminales des arcs en pointillés.

  - Tout nombre premier admet deux diviseurs différents, le nombre 1 et lui-même; donc pour déterminer les nombres premiers, il suffit de déterminer les sommets du graphe dont les pédécesseurs sont 1 et eux-même; c'est à dire $\Gamma^-(x) = \{1, x\}$.

Soit le tableau suivant:

Sommet(x)	L'ensemble des prédécesseurs
1	1
2	1,2
3	1,3
4	1,2,4
5	1,5
6	1,2,3,6
7	1,7
8	1,2,4,8
9	1,3,9

D'après le tableau, l'ensemble des nombres premiers est égal à {2, 3 ,5, 7}.

Sur le graphe les nombres premiers sont les extrémités terminales des arcs en lignes discontinus.

Exercice 02

1
/

Matrices d'adjacence — **Matrice d'incidence**

G1	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	0	1	0
4	0	1	1	0	0	0
5	1	0	0	1	0	0
6	0	0	0	1	1	0

G1 Graphe simple orienté

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10
1	1	0	0	0	0	0	-1	0	0	0
2	-1	1	0	0	-1	1	0	0	0	0
3	0	-1	1	-1	0	0	0	0	0	0
4	0	0	0	1	1	0	0	-1	0	-1
5	0	0	-1	0	0	-1	1	1	-1	0
6	0	0	0	0	0	0	0	0	1	1

G2	1	2	3	4	5	6
1	0	1	0	1	0	1
2	1	0	1	0	0	0
3	0	1	0	1	1	1
4	1	0	1	0	1	0
5	0	0	1	1	0	1
6	1	0	1	0	1	0

G2 Graphe simple non orienté

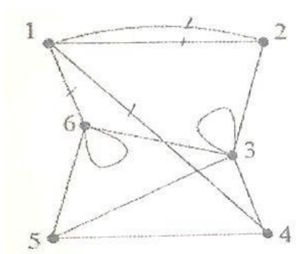
	e1	e2	e3	e4	e5	e6	e7	e8	e9
1	1	1	1	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0
3	0	0	0	1	1	0	1	1	0
4	0	1	0	0	0	0	0	1	1
5	0	0	0	0	0	1	1	0	1
6	1	0	0	0	1	1	0	0	0

G3	1	2	3	4	5	6
1	0	2	0	1	0	1
2	2	0	1	0	0	0
3	0	1	1	1	1	1
4	1	0	1	0	1	0
5	0	0	1	1	0	1
6	1	0	1	0	1	1

G3 Graphe multiple

	1-2	1-2	1-4	1-6	2-3	3-3	3-4	3-5	3-6	4-5	5-6	6-6
1	1	1	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	1	0	0	0	0	0	0	0
3	0	0	0	0	1	2	1	1	1	0	0	0
4	0	0	1	0	0	0	1	0	0	1	0	0
5	0	0	0	0	0	0	0	1	0	1	1	0
6	0	0	0	1	0	0	0	0	1	0	1	2

Exemple de solution pour le graphe G3



Listes des voisins

1	**2**	**2**	**4**	**6**	
2	**1**	**1**	**3**		
3	**3**	**2**	**4**	**5**	**6**
4	**1**	**3**	**5**		
5	**3**	**4**	**6**		
6	**1**	**3**	**5**	**6**	

Les degrés

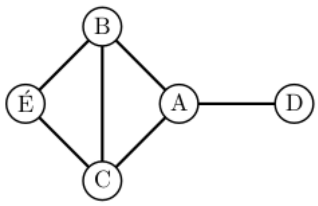
Sommet	Degré
1	**4**
2	**3**
3	**6**
4	**3**
5	**3**
6	**5**

2 /

Représentation	Ajout Arête	Supp. Arête	Ajout sommet	Supp. sommet
Adjacence	Ajouter 1 dans les cellules touchées	1 devient 0 dans les cellules touchées	Ajouter une ligne et une colonne + MAJ	Supprimer une ligne et une colonne + MAJ
Incidence	Ajouter une colonne et MAJ	Supprimer la colonne de l'arête	Ajouter une ligne et des colonnes + MAJ	Supprimer lignes et colonnes + MAJ
Liste des voisins	Ajouter des voisins pour les sommets touchés	Supprimer des voisins dans les sommets touchés	Ajouter une ligne + MAJ	Supprimer les lignes + MAJ

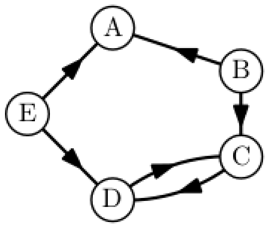
Exercice 03

(a)



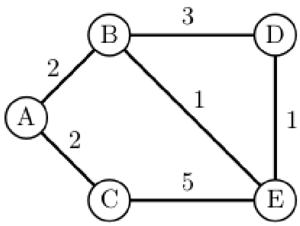
	A	B	C	D	É
A	0	1	1	1	0
B	1	0	1	0	1
C	1	1	0	0	1
D	1	0	0	0	0
É	0	1	1	0	0

(b) En vérifiant Wikipédia, nous constatons que l'article Animal ne renvoie à aucun des autres, l'article Burrow renvoie à Animal et Chili, l'article Chili renvoie à Desert, l'article Desert lie au Chili et l'article Elephant renvoie à Animal et Désert. Étrangement, aucune relation entre Burrow et Elephant ne semble être présente. Nous encodons ces informations avec un graphe orienté.



	A	B	C	D	E
A	0	0	0	0	0
B	1	0	1	0	0
C	0	0	0	1	0
D	0	0	1	0	0
E	1	0	0	1	0

(c)



	A	B	C	D	E
A	0	2	2	0	0
B	2	0	0	3	1
C	2	0	0	0	5
D	0	3	0	0	1
E	0	1	5	1	0

Exercice 04

Chaque état est représenté par un sommet (x,y) indiquant le nombre d'allumettes de chaque tas.

On modélise ce jeu dans le cas où l'on dispose au départ de deux tas contenant chacun trois allumettes, et où un joueur peut enlever une ou deux allumettes à chaque fois :

-Un sommet est un couple de nombres indiquant le nombre d'allumettes restantes dans chaque tas ;

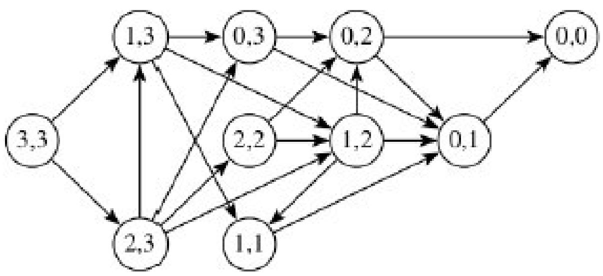
-Une arête indique que l'on peut passer d'une configuration à l'autre. (enlever une ou deux allumettes dans un des tas).

Le jeu avec deux tas de deux allumettes est décrit par le graphe ci-dessus. REM :

-On ne distingue pas les deux tas (par exemple 2,1 = 1,2).

- Notez que la transition entre (2,0) est (0,0) correspond à un suicide, alors qu'on peut jouer un coup gagnant.

Le joueur qui atteint la configuration 0,0 perd la partie. Pour gagner, on doit donc absolument atteindre la configuration 0,1. On peut vérifier qu'en jouant 1,3 au premier coup, on peut toujours ensuite atteindre



0,1 en deux étapes : 1,3 - 0,3 - 0,1
ou 1,3 - 1,2 - 0,1 ou enfin 1,3 - 1,1
- 0,1.

Le coup gagnant au départ est donc « enlever 2 allumettes dans un tas ».

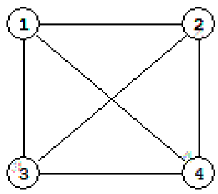
Il faut retirer deux allumettes d’un tas. Quel que soit le coup joué par l’adversaire, on peut jouer un coup gagnant.

Exercice 05

1. On doit prouver que dans un graphe simple (sans arête double, sans boucle), il y a toujours deux sommets qui ont le même degré.
2. On va raisonner par l'absurde et supposer qu'il y a un graphe simple à *n* sommets dont tous les degrés sont différents. Alors les degrés des sommets sont compris entre 0 et $n-1$, et donc si tous les degrés sont différents, comme on a *n* sommets et simplement *n* possibilités pour les degrés, toutes ces possibilités doivent être prises. En particulier, il doit y avoir un sommet de degré 0 et un sommet de degré $n-1$. Mais c'est impossible, car ce dernier sommet devrait avoir une arête vers tous les autres, et ce n'est pas possible puisqu'un sommet est de degré 0, donc isolé.

Exercice 06

1/ Un graphe complet, chaque sommet est relié avec tout les autres sommets.



2/ Si nous avons n sommets dans un graphe complet d(x)=n-1|nbre d’arête= n(n-1)/2

3 / Un graphe simple est un graphe sans arêtes multiples et sans boucles ; On a |U|=(n-1)(n-2)/2 c’est le nbre d’arêtes d’un graphe complet ayant (n-1) sommets (d’après 1 et 2) Comme le graphe est complet et sans boucle => si on ajoute une arête (ou plus , =>rajouter obligatoirement un sommet puisque le graphe est complet) elle doit obligatoirement sortir de la composante connexe qui est le graphe ayant (n-1) sommets pour rejoindre le sommet x (n ème sommet).(le sommet est lié à tous les autres sommets.)

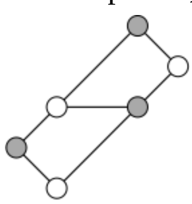
Donc il existe une chaine qui relie x à tous les autres sommets (n-1)=> G possède une seule composante connexe comptant n sommets.

Exercice 07

(a) Ce graphique est planaire, car il n'y a pas de croisements d'arêtes dans son dessin. Il n'est pas bipartite, car il a un cycle de longueur impaire.

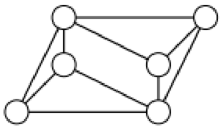
(b) Ce graphe est également planaire: on peut «retourner» une partie du dessin pour obtenir un dessin plan.

Il est également bipartite, car nous pouvons colorer tous les sommets avec deux couleurs.



(c) Ce graphe n'est pas planaire: il a 6 sommets et 14 arêtes, et selon la formule d'Euler un graphe plan à 6 sommets peut avoir au plus 3V - 6 = 12 arêtes. Il n'est pas non plus bipartite, car il contient des triangles.

(d) Ce graphe est également planaire, comme on peut le voir sur ce dessin du même graphique:



Il n'est pas bipartite, car il contient des triangles.

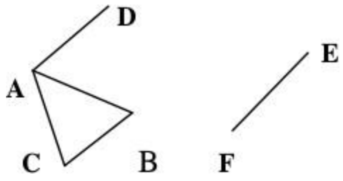
Série de travaux dirigés N° 2

Exercice 01

Trois états envoient chacun à une conférence deux espions. Chaque espion doit espionner tous les espions des autres pays (mais pas son propre collègue !).
1) Représenter cette situation par un graphe. Quel est le degré de chaque sommet ? Déduisez le nombre d'arêtes.
2) Ce graphe est-il complet ?
3) Démontrer le théorème d'Euler en utilisant la notion de la connexité.

Exercice 02

A/
Soit le graphe **G1**

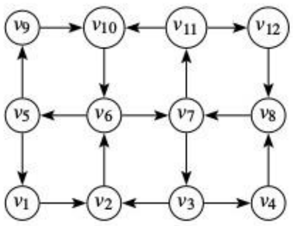


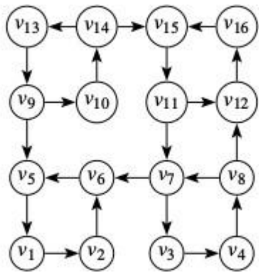
G1

1) Est-il connexe ?
2) Quel est le nombre de composantes connexes?

/B

1) Les graphes ci-contre, sont-il fortement connexes ? Si non, donnez leurs composantes fortement connexes.



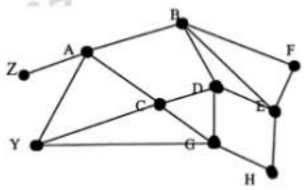


2) Montrer que G (G un graphe quelquonque) est un graphe fortement connexe ssi G est connexe et tout arc est dans un circuit.
3) Est-ce vraie si on remplace la condition précédente par G connexe et tout sommet est dans un circuit.

Exercice 03

Dessinez un graphe d'ordre au moins 5 qui est. . .
1) hamiltonien et eulérien
2) hamiltonien et non eulérien
3) non hamiltonien et eulérien
4) non hamiltonien et non eulérien.

Exercice 04

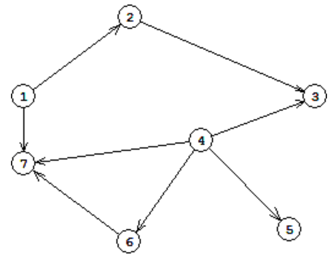


1) On considère le graphe ci-dessus.
a) Ce graphe est-il connexe ?
b) Déterminer le degré de chacun des sommets. On pourra donner le résultat sous forme d'un tableau.
c) Justifier l'existence d'une chaîne eulérienne.
d) Justifier l'absence d'un cycle eulérien.
2) Soit G un graphe non Eulérien. Est-il toujours possible de rendre G eulérien en lui rajoutant un sommet et quelques arêtes ?

Exercice 05

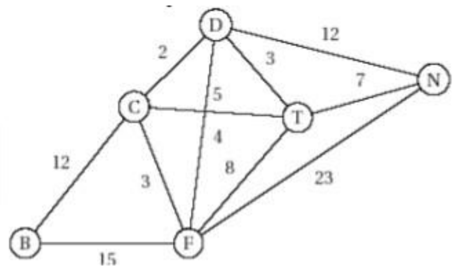
A/

1) Montrer que le graphe G1 ne possède pas de chemins Hamiltonien.



B/

1) Justifier que le graphe G2 est connexe.
2) Le groupe souhaite passer par les six sommets en passant une fois et une seule par chaque chemin.
Démontrer que leur souhait est réalisable. Donner un exemple de trajet possible.
3) Le groupe souhaite associer chaque sommet à une couleur de sorte que les sommets reliés par un chemin n'ont pas la même couleur. On note n le nombre chromatique du graphe.

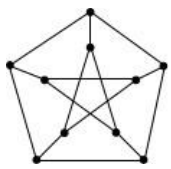


   a) Montrer que $4 \leq n \leq 6$
   b) Proposer un coloriage du graphe permettant de déterminer son nombre chromatique.

4. Est-ce qu'il existe un cycle Hamiltonien, et quelle est sa longueur.

Exercice 06

On cherche à colorier le graphe ci-contre en utilisant des entiers positifs de façon telle que deux sommets voisins ont des couleurs dont la différence, en valeur absolue, est au moins égale à trois.



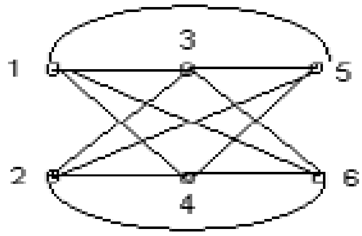
1. Proposez une coloration de ce graphe. Quel est le plus grand entier utilisé ?
2. Peut-on faire mieux ?
3. Maintenant, on souhaite que, de plus, deux sommets à distance deux aient des couleurs dont la différence, en valeur absolue, est au moins égale à deux. Quelle est la meilleure coloration possible de ce graphe ?

Exercice 07

On a six wagons à trier. Dans la gare de triage, les wagons entrent dans l'ordre 2, 5, 3, 6,1, 4 et doivent sortir dans l'ordre croissant. Deux wagons *i* et *j* peuvent être mis sur la même voie si et seulement s'ils entrent dans l'ordre dans lequel ils doivent sortir.
Dessinez un graphe illustrant la situation, en indiquant ce que représentent les sommets et les arêtes de votre graphe. Quel sera le nombre minimal de voies nécessaires au tri ?

Corrigé des travaux dirigés de la série N° 2

Exercice 01



Les espions d'un même pays sont notés 1 et 2 , 3 et 4, 5 et 6
1) Ce graphe n'est pas complet car deux espions d'un même pays ne s'espionnent pas, donc les sommets correspondants ne sont pas adjacents.
En revanche ce graphe est connexe car entre tout couple de points, il existe au moins une chaîne
2) Les sommets sont tous de degré 4 car chaque espion en espionne quatre autres
Autrement dit :

Sommet	1	2	3	4	5	6
Degré	4	4	4	4	4	4

La somme des degrés étant égale au double du nombre d'arêtes, celui-ci vaut 12.

3) Démonstration du théorème d'Euler en utilisant la notion de la connexité.

Démonstration : Supposons que le graphe connexe G possède une chaine eulérienne. Chaque fois qu'en parcourant cette chaine on arrive à un sommet par une arête, on en repart par une autre, sauf pour les extrémités de cette chaine. Comme cette chaine est eulérienne, chaque arête est parcourue une et une seule fois, ainsi si la chaine est passée n fois par le sommet s, ce sommet est de degré 2n. Par conséquent tous les sommets du graphe sont pairs sauf peut-être deux (les extrémités de la chaine si elles sont différentes).
Nous allons démontrer réciproquement que, si tous les sommets d'un graphe connexe sont pairs sauf deux (notés u et v), il existe une chaine eulérienne reliant ces deux sommets, ceci par récurrence sur le nombre d'arêtes du graphe. L'on en déduira alors qu'un graphe connexe dont tous les sommets sont pairs possède un cycle eulérien. En effet, si dans un tel graphe on ôte arbitrairement une arête a, on est ramené au cas précédent, la chaine eulérienne est alors complétée en un cycle eulérien par ajout de l'arête a. On peut toujours le faire car il n'y a pas de sommet de degré zéro dans un graphe connexe.

Exercice 02
A/
1/ Un graphe est connexe

$$\forall\, x, y \in X \begin{cases} x = y \\ \exists \text{ une chaine reliant } x \text{ à } y \end{cases}$$

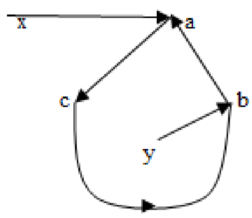
Le graphe g n'est pas connexe car il n'existe pas une chaine reliant tout les sommets (il n'existe pas une chaine avec E et F)
2/ il existe deux composantes connexes
C1 ={A,B,C,D}   C2 ={E,F}

3/ Pour rendre le graphe connexe , il faut relier les deux composantes connexes par une arête ayant une extrémité dans C1 et une deuxième extrémité dans C2.
B/
1. Rappel connexité forte:

$$\forall\, x, y \in X \begin{cases} x = y \\ \exists \text{ un chemin reliant } x \text{ à } y \text{ et un chemin de } y \text{ à } x \end{cases}$$

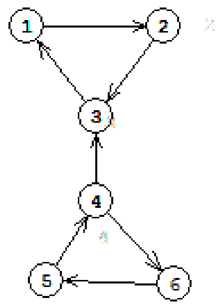
1) Le premier graphe est fortement connexe, car tous les sommets ont pu être marqués "±". Par contre, le deuxième digraphe n’est pas fortement connexe. Pour trouver les composantes fortement connexes, on suit la procédure suivante :
a. Appliquer l’algorithme de marquage.
b. Supprimer les sommets marqués d’un "±" : ils forment une composante fortement connexe.
c. Tant qu’il reste des sommets, aller en a.
Composantes fortement connexes du deuxième digraphe : {1, 2, 5, 6}, {9, 10, 13, 14}, et {3, 4, 7, 8, 11, 12, 15, 16}.
2) Appliquer l'
GFC ⎕ 1) G est connexe
2) tout arc est dans un circuit

GFC => G est connexe , tout arc appartient à un circuit.
Soit un arc (x,y) puisque G est fortement connexe alors ∃ un chemin de y à x, donc si on ajoute à la fin de ce chemin l’arc (x,y) on obtient un circuit qui contient l’arc (x,y).
Gc et tout arc est dans un circuit => GFC



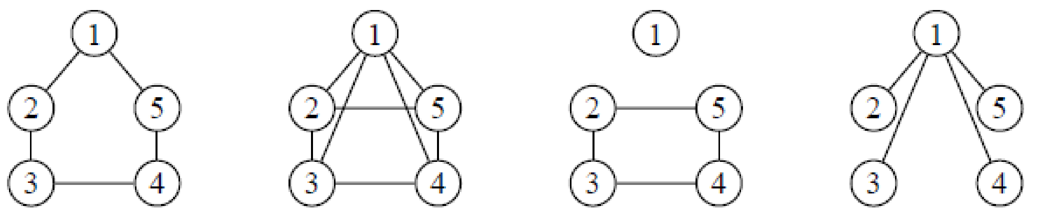
1. On supposera que G est fortement connexe. Par définition, G est connexe. Soit (x; y) un arc de G. Comme G est fortement connexe, il existe un chemin de y vers x. Si on ajoute à la fin de ce chemin l’arc (x; y) on obtient un circuit contenant x et y.

Réciproquement, supposons que G est connexe et que tout arc est dans un circuit. Soient
x ,y ∈ V . Il existe une chaîne de x vers y. Supposons qu’il existe dans cette chaine un arc (a; b) "dans le mauvais sens". Comme (a; b) appartient à un circuit, il existe un chemin C de b vers a. On remplace alors (a, b) dans le chemin non-orienté de x vers y par le chemin C de b ⎕a. On applique cette transformation à tous les arcs qui sont dans le mauvais sens. On obtient un chemin de x vers y. G est donc fortement connexe



3/ Non: Un contre exemple:

Exercice 03



Hamiltonien et eulérien | Hamiltonien et non eulérien | Non hamiltonien et eulérien | Non hamiltonien et non eulérien

Exercice 04

1. a) Le graphe est connexe car entre tout couple de sommets, il existe au moins une chaîne.
b) Le tableau donnant les degrés de chaque sommet est :

Sommet	A	B	C	D	E	F	G	H	Y	Z
Degré	4	4	4	4	4	2	4	2	3	1

c) Puisque seuls les deux sommets Y et Z sont de degré impair, le théorème d'Euler affirme l'existence d'une chaîne eulérienne.

d) Puisque y'a un sommet de degré impair, le théorème d'Euler inffirme l'existence d'un cycle eulérienne.

2 . Pour qu'un graphe soit eulérien, il faut et il suffit que tous ses sommets soient de degré pair. Si un graphe contient k sommets impairs, il est possible de rajouter un nouveau sommet x, relié à ces k sommets. Dans le graphe obtenu, les k sommets considérés sont devenus pairs… Cependant, le degré de x étant k, le graphe n'est toujours pas eulérien si k était impair…
Remarquons qu'il est possible de rajouter des arêtes entre les sommets de degré impair dans le graphe d'origine… Mais l'ajout d'une telle arête, entre deux sommets impairs a et b par exemple, fait que le nombre de sommets impairs devient k-2, qui a la même parité que k…
La réponse est donc :si C'est possible…

Exercice 05
A / Chemin Hamiltonien : il faut passer une seule fois par tout les sommets. Le graphe ne possède pas un chemins Hamiltonien car : $d^+_G(3) = d^+_G(5)=0$ (pas d'arcs sortants) => les sommets 3 et 5 ne peuvent appartenir au chemin Hamiltonien.
B /
1/ Le graphe est connexe

$$\forall\, x, y \in X \begin{cases} x = y \\ \exists\ une\ chaine\ reliant\ x\ à\ y \end{cases}$$

2/ a) Justifier que le graphe est connexe.
Ce graphe est connexe car tous les sommets peuvent être reliés entre eux par (au moins) une chaine.
Par exemple, la chaîne BCDNTF contient tous les sommets.
2) L'existence d'un parcours permettant au groupe de passer par les six sommets en passant une fois et une seule par
chaque chemin est liée à l'existence d'une chaîne eulérienne.
Puisque deux sommets exactement sont de degré impair et que les autres sont de degré pair, le théorème d'euler nous
permet d'affirmer l'existence d'une telle chaîne eulérienne, donc d'un tel parcours.
Par exemple, le trajet F-B-C-F-N-T-F-D-C-T-D-N répond au problème.
3) a) Le sommet ayant le plus grand degré est le sommet F, de degré 5.
Le cours nous affirme qu'alors $n \leq 5 + 1$, c'est-à-dire $n \leq 6$.
De plus, le sous-graphe FCTD, d'ordre 4, étant complet, on aura $n \geq 4$ (il faudra au moins 4 couleurs pour le colorier).
b) On utilise l'algorithme de coloration pour colorier le graphe:

Sommet	Degré	Couleur
F	5	Couleur 1
C	4	Couleur 2
D	4	Couleur 3
T	4	Couleur 4
N	3	Couleur 2
B	2	Couleur 4

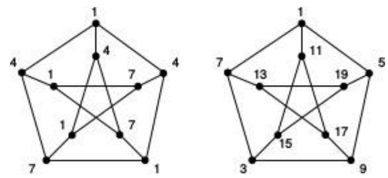
Le nombre chromatique de ce graphe est donc égal à 4.

Exercice 06

Voici deux colorations du graphe de Petersen. La première utilise 7 comme plus grand entier, la deuxième 19...

Dans le premier cas, on peut vérifier que pour colorier ainsi un cycle à 5 sommets, la meilleure solution est 1-4-1-4-7. Ainsi, il n’est pas possible de n’utiliser que des entiers inférieurs à 7.

Dans le deuxième cas, remarquons que deux sommets quelconques sont à distance au plus 2 (on dit que le graphe est de diamètre 2). Toutes les couleurs doivent donc être distinctes et « espacées » de 2. La meilleure solution utilise ainsi les entiers impairs 1,3,5,...,19.



Exercice 07

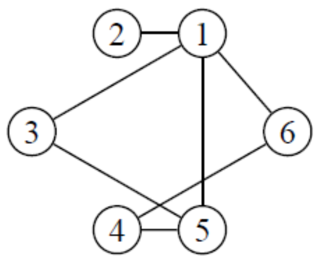
On représente les wagons par les sommets. Une arête relie deux sommets *i* et *j* si les wagons *i* et *j* ne peuvent pas être sur la même voie.

Gérer les incompatibilité par graphe d’incomptibilité

On obtient le graphe ci-contre.

On voit que 1, 3 et 5 ne peuvent pas être sur la même voie.

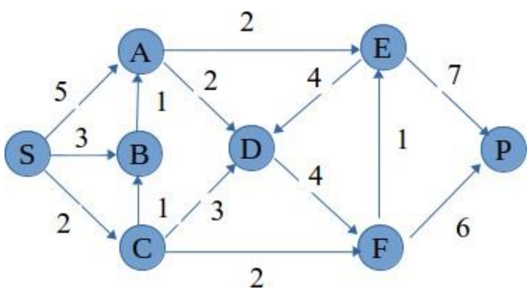
Il faut donc trois voies au minimum.



Série de travaux dirigés N° 3
Chemin le plus court, Flots & Arbres couvrants de poids minimum

Partie 01: Chemin le plus court

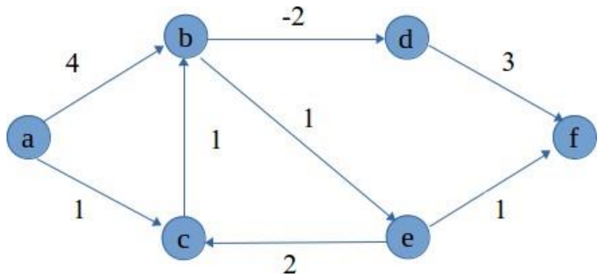
Exercice 01



On veut partir du point A pour se rendre au point F.

Déterminer le chemin qu'on doit suivre pour que le temps de parcours soit le plus court possible. Donner ce temps de parcours.

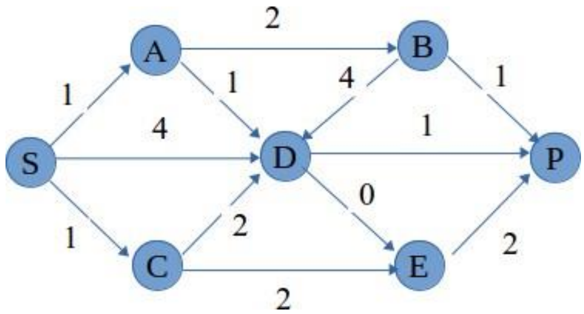
Exercice 02



1) Déterminer, dans le graphe suivant, les plus courts chemins à partir du sommet a au sommet f. Utiliser l'algorithme adéquat (préciser pourquoi cela est nécessaire).

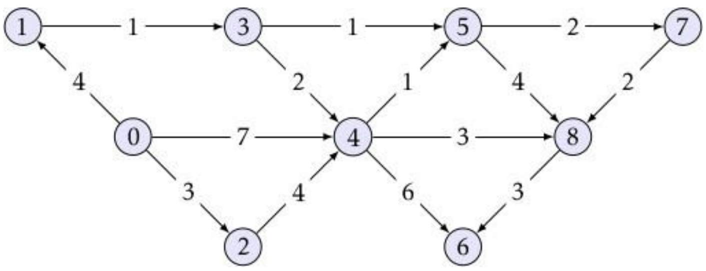
2) Expliquez pourquoi des arcs avec des poids négatifs pourraient poser problème dans la recherche d'un plus court chemin dans un graphe.

Exercice 03



Choisir un algorithme (vu au cours) et l'appliquer pour trouver le plus long chemin allant du sommet S au sommet P dans le graphe suivant.

Exercice 04

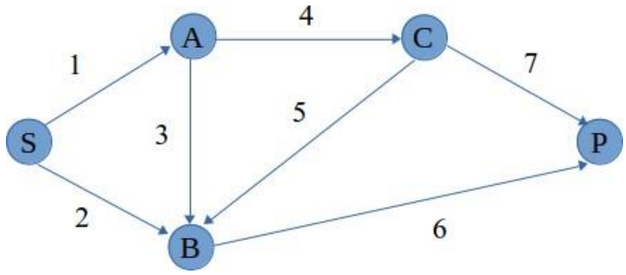


Dans le graphe suivant, il existe de nombreux chemins de poids minimal menant du sommet 0 au sommet 8, mais lequel est retourné par l'algorithme de Dijkstra ? (on conviendra qu'à chaque itération, le chemin optimal menant à un sommet v n'est mis à jour qu'en cas de découverte d'un chemin de poids strictement inférieur).

Partie 02: Flot maximal

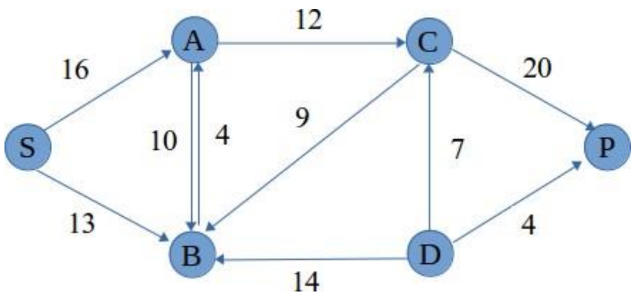
Exercice 01

Donner toutes les coupes entre les 2 sommets S et P dans le graphe suivant:



Exercice 02

Déterminer le flot maximum que l'on peut faire passer à travers ce réseau.



Exercice 03

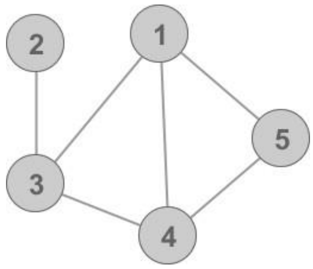
On dispose de trois entrepôts A, B et C qui contiennent respectivement 30 , 20 et 45 tonnes de marchandise, ainsi que cinq destinations D, E, F, G et H qui peuvent faire passer les quantités 10, 25, 20, 25 et 15 tonnes. Le tableau suivant montre les possibilités de transport:

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

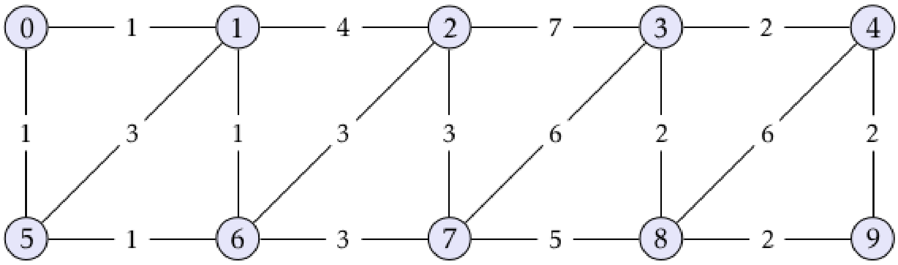
Trouver la quantité maximale de marchandise que l'on peut faire passer à travers ce réseau.

Partie 03: Arbres couvrant de poids minimal

Exercice 01

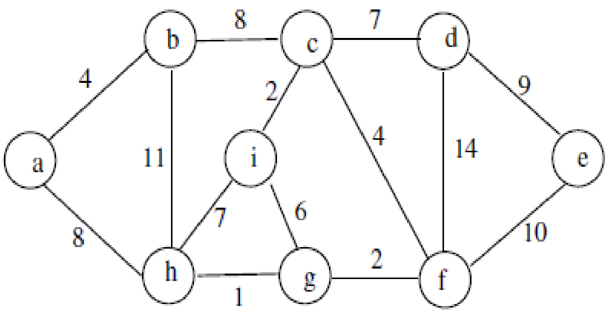


1) Combien d'arbres couvrants différents et comportant obligatoirement l'arête (1,4), le graphe G ci-contre possède-t-il?
2) Appliquer les algorithmes de Prim et de Kruskal au graphe ci-dessous.



Exercice 02

On désire alimenter une ville en eau potable. Seulement, Cette opération nécessite l'installation d'une canalisation d'eau entre les différents quartiers de la ville modélisée par le graphe à côté:



On veut réaliser ce projet de telle sorte que le coût total soit minimum.

2) Les cycles sont ils souhaitables dans le futur graphe? Dîtes pourquoi?
3) Montrer que la résolution de ce problème revient à extraire de G un graphe partiel G', Déduire la nature de G';
4) Appliquer l'algorithme de KRUSKAL au graphe G donné.

Exercice 03

Montrer qu'un graphe G contient un arbre couvrant si et seulement si G est connexe.

Exercice 04

Montrer qu'un graphe G sans boucle est un arbre si et seulement si, pour toute paire x, y de sommets distincts de G, il existe une et une seule (x, y)-chaîne élémentaire dans G.

Corrigé des travaux dirigés de la série N° 3

Exercice 01

On applique l'algorithme de Djikstra (Bellman-Ford est applicable aussi).

Sommet/ Itération	S	A	B	C	D	F	G	P	Sommet fixé	L_{min}
	0	∞	∞	∞	∞	∞	∞	∞	S	0
		5 (S)	3 (S)	**2(S)**	∞	∞	∞	∞	C	2
		5 (E)	**3 (E)**		∞	4 (C)	5 (C)	∞	B	3
		4 (B)			∞	4 (C)	5 (C)	∞	A	4
					6 (A)	**4 (C)**	5 (C)	∞	E	4
					5(F)		5 (C)	10 (F)	D	5
							5 (C)	10 (F)	F	5
								10 (F)	**P**	10

Pour déterminer le trajet le plus court on remonte les sommets en partant de S : S vient de F qui vient de C qui vient de E.

> Le plus court chemin est S-C-F-P, la distance parcourue est de 10 km.

Exercice 02

Le graphe possède un circuit et des arcs de longueur négative.
On peut appliquer l'algorithme de Ford-Bellman

	a	b	c	d	e	f
Initialisation	0	+∞	+∞	+∞	+∞	+∞
Itération1	//	4(a)	1(a)	2(b)	5(b)	5(d)
Itération 2	//	2(c)	--	0(b)	3(b)	3(d)
Itération 3	//	--	--	--	--	--

A chaque itération on passe en revue tous les sommets et on compare leur marque à celle de leurs prédécesseurs. On s'arrête lorsqu'aucune marque n'est modifiée.

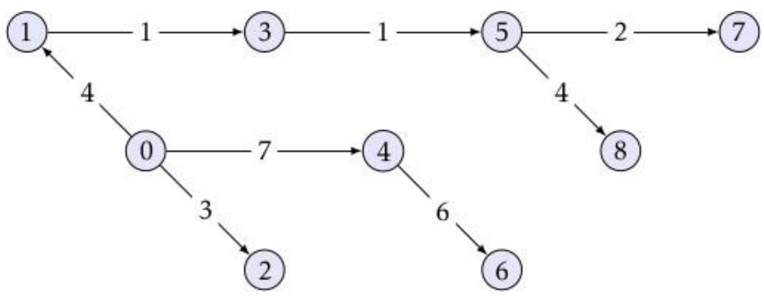
Exercice 03:
Le chemin le plus long.

Exercice 04:

Suivons l'évolution de l'algorithme de Dijkstra dans le tableau ci-dessous:

S	0	1	2	3	4	5	6	7	8
$\{0\}$	·	4	3	$+\infty$	7	$+\infty$	$+\infty$	$+\infty$	$+\infty$
$\{0,2\}$	·	4	·	$+\infty$	7	$+\infty$	$+\infty$	$+\infty$	$+\infty$
$\{0,2,1\}$	·	·	·	5	7	$+\infty$	$+\infty$	$+\infty$	$+\infty$
$\{0,2,1,3\}$	·	·	·	·	7	6	$+\infty$	$+\infty$	$+\infty$
$\{0,2,1,3,5\}$	·	·	·	·	7	·	$+\infty$	8	10
$\{0,2,1,3,5,4\}$	·	·	·	·	·	·	13	8	10
$\{0,2,1,3,5,4,7\}$	·	·	·	·	·	·	13	·	10
$\{0,2,1,3,5,4,7,8\}$	·	·	·	·	·	·	13	·	·
$\{0,2,1,3,5,4,7,8,6\}$	·	·	·	·	·	·	·	·	·

Dans le même temps, on mémorise les prédécesseurs de chaque sommet dans un chemin optimal, ce qu'on peut représenter sur le graphe:



Le chemin retourné par l'algorithme de Dijkstra est donc : (0, 1, 3, 5, 8).

Série de travaux dirigés N° 4 (Programmation Linéaire)

Exercice 01

Un verrier produit des verres en trois modèles différents. Les prix de vente, les quantités requises de verre ainsi que les temps de façonnage et d'emballage sont différents pour chacun des produits et sont résumés dans le tableau suivant :

Taille	Modèle 1	Modèle 2	Modèle 3
Temps de façonnage (min)	4	2	12
Temps d'emballage (min)	2	1	4
Quantité de verre (kg)	0.1	0.15	0.1
Prix de vente (DZD)	80	60	150

Pour le mois à venir, l'artisan dispose de 50 heures pour le façonnage, 20 heures pour l'emballage et de 100 kilogrammes de verre.

1 – Déterminer les variables de décision de ce problème d'optimisation qui serviront à proposer une production visant à maximiser le chiffre d'affaire de ce verrier.
2 – Formuler cette problématique sous forme d'un programme linéaire sous les deux formes canonique et standard.

Exercice 02

Une usine fabrique deux types de jouets en bois : des soldats et des trains. Les données de ce problème sont représentées dans le tableau suivant :

	P. vente	Mat. prem.	Frais gén.	Menuiserie	Finition
1 soldat	27 u	10 u	14 u	1h de travail	2h de travail
1 train	21 u	9 u	10 u	1h de travail	1h de travail

Par semaine l'usine dispose de toutes les matières premières nécessaires à la fabrication et ne dispose que de 100h de finition et 80h de menuiserie.
La demande des trains et des soldats est illimitée. Déterminer le plan de production qui maximise le profit de l'usine.

Exercice 03

Pour fabriquer deux produits P1 et P2, on doit effectuer des opérations sur trois machines M1, M2 et M3, successivement mais dans un ordre quelconque. Les temps unitaires d'exécution sont donnés par le tableau suivant:

	M1	M2	M3
P1	11 mn	7 mn	6 mn
P2	9 mn	12 mn	16 mn

On supposera que les machines n'ont pas de temps d'inactivité.

La disponibilité pour chaque machine sont :

165 heures (9900 minutes) pour la machine M1;

140 heures (8400 minutes) pour la machine M2;

160 heures (9600 minutes) pour la machine M3.

Le produit P1 donne un profit unitaire de 900 dinars et le produit P2 un profit unitaire de 1000 dinars. Dans ces conditions, combien doit-on fabriquer mensuellement de produits P1 et P2 pour avoir un profit total maximum?

Exercice 04

Un fabricant produit 2 types de yaourts à la fraise A et B à partir de fraise, lait et de Sucre. Chaque yaourt doit respecter les proportions suivantes de matières premières:

	Fraise	Lait	Sucre
A	2	1	0
B	1	2	1

Les matières premières sont en quantité limitée : 800 kilos de Fraises, 700 kilos de Lait et 300 kilos de sucre. La vente des yaourts A rapportent 4 unités monétaires par kilo et les yaourts B 5 unités monétaires. Comment avoir un profit maximum ? (Résolution algébrique)

Exercice 05

On souhaite tirer le meilleur rendement d'un avion transporteur , qui rapporte 3K euros par tonne de fret transportée dans la cabine et 1K euros par tonne de fret transportée dans la soute, sachant que la capacité de la soute est de 20 tonnes et celle de la cabine est de 10 tonnes. Pour des raisons de sécurité, la charge maximale que peut accepter l'avion est de 28 tonnes et enfin, pour des raisons d'équilibrage, le fret de la cabine amputé d'une tonne ne doit pas excéder les deux tiers du fret de la soute.

1. Modéliser ce problème comme un programme linéaire sous forme canonique.
2. Effectuer une résolution graphique et en déduire le rendement optimal par vol.
3. Mettre le programme de la Question 1 sous forme standard et le résoudre par la méthode du simplexe tableau.

Corrigé des travaux dirigés de la série N° 4

Exercice 1:

1 - Les variables de décision:
X1 : Quantité de verres du Modèle 1 à fabriquer pendant le mois à venir.
X2 : Quantité de verres du Modèle 2 à fabriquer pendant le mois à venir.
X3 : Quantité de verres du Modèle 3 à fabriquer pendant le mois à venir.

2 – **Forme canonique :**
Max $Z = 80x_1 + 60x_2+150x_3$
S.c.: $4x_1+2x_2+12x_3 <=3000$
$2x_1+x_2+4x_3 <= 1200$
$0.1x_1 + 0.15\, x_2 + 0.1\, x_3 <= 100$
$x_i>=0$, i{1,2,3}.

Forme standard :
Max $Z = 80x_1 + 60x_2+150x_3$
S.c.: $4x_1+2x_2+12x_3 + e_1 = 3000$
$2x_1+x_2+4x_3 + e_2 = 1200$
$0.1x_1 + 0.15\, x_2 + 0.1\, x_3 + e_3 = 100$
$x_i>=0$, i{1,2,3}.
$e_i>=0$, i{1,2,3}.

Exercice 2 :

On note par :

x_1 = le nombre de soldats produits chaque semaine ,

x_2 = le nombre de trains produits chaque semaine.

D'après les données, on a les contraintes suivantes :

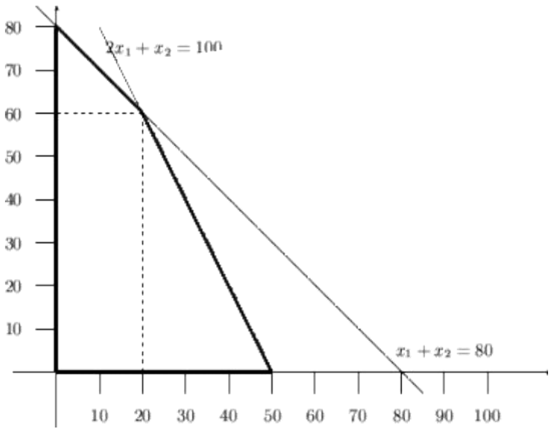
D'après les données, on a les contraintes suivantes $\begin{cases} x_1 + x_2 & \leq 80 \\ 2x_1 + x_2 & \leq 100 \\ x_1 \geq 0, x_2 & \geq 0 \end{cases}$

La fonction à maximiser est la suivante : $(27 - 10 - 14)x_1 + (21 - 9 - 10)x_2$

Ce qui donne le programme linéaire suivant $\begin{cases} \max\ 3x_1 + 2x_2 \\ x_1 + x_2 \leq 80 \\ 2x_1 + x_2 \leq 100 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$

Le programme est de dimension deux, donc,

on le résout graphiquement :



Exercice 02

Formulation en un PL:

Les variables de décisions sont:

x_1 : le nombre d'unités du produit P1 à fabriquer

x_2 : le nombre d'unités du produit P2 à fabriquer

Les contraintes outre les contraintes de non-négativité sont:

$11\, x_1 + 9\, x_2 \leq 9900$ pour la machine M1

$7\, x_1 + 12\, x_2 \leq 8400$ pour la machine M2

$6\, x_1 + 16\, x_2 \leq 9600$ pour la machine M3

Le profit à maximiser est: $z = 900\, x_1 + 1000\, x_2$

Le programme linéaire résultant est:

$$\begin{array}{ll} \text{Max} & 900\, x_1 + 1000\, x_2 \\ s.c. & 11\, x_1 + 9\, x_2 \leq 9900 \\ & 7\, x_1 + 12\, x_2 \leq 8400 \\ & 6\, x_1 + 16\, x_2 \leq 9600 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Problème à deux variable Résolution graphique

Faite le graphe

Exercice 4

Modélisation par forme canonique:

Max Z= $4\, x_1 + 5\, x_2$

$2\, x1 + x2 \leq 800$

$x1 + 2\, x2 \leq 700$

-La forme standard:

Max Z= $4\, x_1 + 5\, x_2$

$2\, x_1 + x_2 + e_1 + 0e_2 + 0\, e_3 = 800$

$x_1 + 2\, x_2 + 0\, e_1 + e_2 + 0\, e_3 = 700$

$x_2 + 0\, e_1 + 0\, e_2 + e3 = 300$

x_1 , x_2 ³ 0

$x2 \leq 300$
$x1 , x2 \geq 0$

1ière itération point =(0,0)

Base	x_1	x_2	e_1	e_2	e_3	C	K
e_1	2	1	1	0	0	800	800
e_2	1	2	0	1	0	700	350
e_3	0	1	0	0	1	300	300
(-Z)	4	5	0	0	0	0	

2ième itération

Base	x_1	x_2	e_1	e_2	e_3	C	K
e_1	2	0	1	0	-1	500	250
e_2	1	0	0	1	-2	100	100
x_2	0	1	0	0	1	300	300
(-Z)	4	0	0	0	-5	-1500	

Ce tableau correspond au point A=(0,300)

3ème itération

Base	x_1	x_2	e_1	e_2	e_3	C	K
e_1	0	0	1	-2	3	300	100
x_1	1	0	0	1	-2	100	-50
x_2	0	1	0	0	1	300	300
(-Z)	0	0	0	-4	3	-1900	

Ce tableau correspond au point A=(100,300)

4ème itération

Base	x_1	x_2	e_1	e_2	e_3	C	K
e_1	0	0	1/3	-2/3	1	100	
x_1	1	0	2/3	-1/3	0	300	
x_2	0	1	-1/3	2/3	0	200	
(-Z)	0	0	-1	-2	0	-2200	

Les coefficients $d_{ij} \leq 0$. Terminer , la solution est optimale.
Cette solution correspond au point P = (200,300) avec un profit maximum de 2200.
Pour 200 yaourt de type A et 300 yaourt de type B le fabricant reçoit un profit maximum de 2200 unités monétaires.

Exercice 5

Les variable de décisions:

X1: la quantité de fret transporté en soute.
X2: la quantité de fret transporté en cabine.

Les contraintes:

- **Capacité de la soute:** $X_1 <= 20$
- **Capacité de la cabine:** $X_2 <= 10$
- **Charge maximale:** $X_1 + X_2 <= 28$
- **Equilibrage:** $X2 - 1 \leq 2/3\ X1$
- $X1 \geq 0$
- $X2 \geq 0$

La fonction objectif: Max Z = X1 + 3X2
La solution graphique:

(0,0) => p = 0

(0,1) => p = 3

(5,10) => p = 35

(18,10) => p = 48

(20, 8) => p = 44

(20,0) => p = 20

La solution simplex:

Modélisation par forme canonique:

Max Z = X1 + 3X2
$X_1 \leq 20$
$X_2 \leq 10$
$X_1 + X_2 \leq 28$
$X_2 - 1 \leq 2/3\ X1$

$X_1, X_2 >= 0$

-La forme standard :

Max Z= $X_1 + 3X_2$

$x_1 + 1e_1 + 0e_2 + 0e_3 + 0e_4 = 20$
$x_2 + 0\ e_1 + 1e_2 + 0\ e_3 + 0e_4 = 10$
$x_1 + x_2 + 0\ e_1 + 0\ e_2 + 1e_3 + 0e_4 = 28$
$-2/3x_1 + x_2 + 0\ e_1 + 0\ e_2 + 0\ e_3 + 1e_4 = 1$

$x_1, x_2 \geq 0$

1ière itération point =(0,0)

Base	x_1	x_2	e_1	e_2	e_3	e_4	C	K
e_1	1	0	1	0	0	0	20	800
e_2	0	1	0	1	0	0	10	350
e_3	0	1	0	0	1	0	28	300
e_4	-2/3	1	0	0	0	0	1	
(-Z)	0	3	0	0	0	0	0	

2ième itération

Base	x_1	x_2	e_1	e_2	e_3	e_4	C	K
e_1	1	0	1	0	0	0	20	800
e_2	0	1	0	1	0	0	10	350
e_3	0	1	0	0	1	0	28	300
e_4	-2/3	1	0	0	0	0	1	
(-Z)	0	3	0	0	0	0	0	

Bibliographie

MÜLLER, Didier. *Introduction à la théorie des graphes*. Commission romande de mathématique, 2011.

ROUX, Christian. *Initiation à la théorie des graphes*. Ellipses, 2009.

ANDJIGA, N. G. INITIATION À LA THÉORIE DES GRAPHES POUR LYCÉES ET COLLEGES.