

RECHERCHE OPÉRATIONNELLE

Dr. Tchalla Ayékotan Messan Joseph

Enseignant-chercheur à l'Université de Lomé

TABLE DES MATIÈRES

1	Programmation Linéaire	7
1.1	Exemple de programme linéaire	7
1.2	Forme générale, formes matricielles d'un programme linéaire	8
1.2.1	Forme canonique	10
1.2.2	Forme standard des problèmes	10
1.2.3	Réduction à la forme standard d'un problème de programmation linéaire	10
1.3	Exemples de modélisation de quelques problèmes de PL	11
1.4	Résolution par la méthode graphique d'un programme linéaire dépendant de deux ou trois variables	13
1.5	La géométrie de la programmation linéaire	16
1.5.1	polyèdre et ensembles convexes	16
1.6	Bases ou matrices Bases-Solutions de base- Solutions réalisable de base	18
1.7	Polyèdre en forme standard	18
1.7.1	Définitions	18
1.7.2	Procédure pour la construction des solutions de base d'un polyèdre sous forme standard	19
1.7.3	Cas où le rang de la matrice A est inférieur au nombre de lignes de A.	23
1.8	Dégénérescence	24
1.9	Existence des points extrêmes	27
1.10	Optimalité des points extrêmes	27
1.11	La méthode du Simplexe	27
1.11.1	Conditions d'optimalité	28
1.11.2	Développement de l'algorithme du simplexe	35
1.11.3	Une itération de l'algorithme du simplexe	37
1.11.4	Remarques sur la méthode du simplexe pour les problème dégénérés	38
1.11.5	Sélection du pivot	38
1.11.6	Algorithme du simplexe révisé	39
1.11.7	Une itération de la méthode du simplexe révisée	40
1.11.8	Implémentation de l'algorithme du simplexe dans un tableau.	40

1.11.9	Itération de l'algorithme du simplexe dans un tableau	41
1.11.10	Exemple de l'algorithme du simplexe pour les problème dégénérés	46
1.11.11	Méthode du simplexe en deux phases	46
1.12	Dualité en programmation linéaire	48
1.12.1	Motivation	48
1.12.2	problème dual	50
1.12.3	Forme générale d'un problème dual	50
1.12.4	théorèmes fondamentaux de la dualité	51
1.12.5	Relation entre le primal et le dual	51
1.12.6	Écarts complémentaires	52
2	Théorie des graphes	53
2.1	Définition d'un graphe	53
2.2	Graphe non orienté	53
2.2.1	Définitions	53
2.2.2	Représentation graphique d'un graphe non orienté	53
2.2.3	Quelques types de Graphes	54
2.3	Graphe orienté	55
2.3.1	Définitions	55
2.3.2	Représentation graphique d'un graphe orienté	56
2.4	Degré d'un sommet-degré d'un graphe	56
2.5	Chemin-circuit	57
2.6	Graphe partiel et sous graphe	58
2.7	Coloration des sommets d'un graphe	59
2.7.1	Stable d'un graphe	59
2.7.2	Définition de la coloration des sommets d'un graphe	60
2.7.3	Encadrement du nombre chromatique	60
2.7.4	Algorithme glouton	61
2.7.5	Algorithme de Welsh-Powell	61
2.8	Représentation non graphique d'un graphe	62
2.8.1	Représentation non graphique d'un graphe non orienté	62
2.8.2	Représentation non graphique de digraphe	63
2.9	Détermination des plus courts chemins	64
3	Flot dans les graphes	71
3.1	Définitions	71
3.2	Problème du flot maximum	74
3.2.1	Algorithme de Ford et Fulkerson	74

RECHERCHE OPÉRATIONNELLE : INTRODUCTION GÉNÉRALE

La recherche opérationnelle (RO) est une branche des mathématiques appliquées qui donne des outils ou des méthodes permettant de résoudre les questions d'optimisation de grandeurs sous certaines conditions ou contraintes. Optimiser une grandeur c'est trouver le minimum ou le maximum que cette grandeur peut atteindre. La valeur maximale ou minimale de la grandeur est appelée sa valeur optimale.

La RO est utilisée dans les domaines comme l'industrie, l'économie, la finance, le marketing et la planification d'entreprise... Elle aide à de meilleures prises de décisions c'est-à-dire qu'elle aide à faire de bons choix dans ces domaines.

La recherche opérationnelle est née pendant la seconde guerre mondiale des efforts conjugués d'éminents mathématiciens (dont von Neumann, Dantzig, Blackett) à qui il avait été demandé de fournir des techniques d'optimisation des ressources militaires. Le premier succès de cette approche a été obtenu en 1940 par le Prix Nobel de physique Patrick Blackett qui résolut un problème d'implantation optimale de radars de surveillance. Le qualificatif "opérationnel" vient du fait que les premières applications de cette discipline avaient trait aux opérations militaires. Après la guerre, les techniques de RO se sont considérablement développées grâce, notamment, à l'explosion des outils informatiques qui étaient dès lors à la hauteur des méthodes proposées par la RO.

Quelques exemples d'application de la RO

- Planifier la tournée d'un véhicule de livraison qui doit passer par les points fixés à l'avance puis revenir à son point de départ en cherchant à minimiser la distance parcourue est un problème typique de RO. On appelle ce problème **le problème du voyageur de commerce**.
- Remplir un conteneur avec des objets de tailles et de valeurs variables. Si le conteneur a une capacité finie, on peut chercher à maximiser la valeur placée dans le conteneur. On appelle ce problème, le problème du sac-à-dos.

Objectif de cours

L'objectif de ce cours de donner quelques outils et méthodes (algorithmes) permettant de résoudre certains problèmes de recherche opérationnelle.

CHAPITRE 1

PROGRAMMATION LINÉAIRE

Introduction

Dans ce chapitre, nous introduisons la programmation linéaire qui est un outil très puissant de la R.O.

La programmation linéaire ou un programme linéaire est un problème de recherche opérationnelle qui a pour but de maximiser ou de minimiser une fonction réelle linéaire (objectif ou fonction objectif) de plusieurs variables sous un certain nombre de contraintes linéaires qui lient ces variables. ces variables sont appelées **variables de décision**. Les contraintes sont souvent exprimées en termes d'inéquations ou d'équations. En dimension deux (2), c'est-à-dire lorsque le problème est un programme linéaire dépendant de deux variables de décision, le problème peut être résolu aussi bien par une méthode géométrique ou méthode graphique que par une méthode algébrique. Dans les cas de dimension supérieure ou égale à 3, le programme trouve sa résolution, de manière exacte, par une méthode algébrique. La méthode algébrique la plus connue pour résoudre les programmes linéaires en nombres réels est appelée la **la méthode du Simplexe**. Mais la résolution d'un programme linéaire par la méthode du simplexe passe d'abord par la transformation dudit programme en un autre programme linéaire équivalent dit la forme standard du programme linéaire.

1.1 Exemple de programme linéaire

Les exemples suivants sont des problèmes de programmation linéaire

Exemple 1.

$$\begin{array}{llllll} \text{minimiser} & 2x_1 - & x_2 + & 4x_3 & & \\ \text{sous les contraintes} & x_1 + & x_2 + & & x_4 & \leq 2 \\ & & 3x_2 - & x_3 & & = 5 \\ & & & x_3 & + x_4 & \geq 3 \\ & & & x_1 & & \geq 0 \\ & & & & x_3 & \leq 0. \end{array}$$

Exemple 2.

$$\begin{array}{ll} \text{maximiser} & 400x + 500y \\ \text{sous les contraintes} & 2x + y \leq 800 \\ & x + 2y \leq 700 \\ & y \leq 300 \\ & x \geq 0 \\ & y \geq 0. \end{array}$$

1.2 Forme générale, formes matricielles d'un programme linéaire

De façon générale, dans un problème de programmation linéaire, il est donné un vecteur $c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$ appelé **vecteur coût** et l'objectif est de minimiser ou de maximiser une fonction linéaire

$c^T x = \sum_{j=1}^n c_j x_j$ appelée fonction coût (ou fonction économique ou fonction objectif ou critère) dans

un ensemble de vecteurs $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ soumis à un ensemble de contraintes linéaires exprimées

par des équations et/ou des inéquations. En particulier, soient M_1 , M_2 et M_3 des ensemble finis d'indices. Supposons que pour tout i appartenant à un quelconque de ces ensembles, il est donné

un vecteur $a_i = \begin{pmatrix} a_{i1} \\ \vdots \\ a_{in} \end{pmatrix}$ et un réel b_i pour former la i -ème contrainte. Soient aussi N_1 et N_2 deux

sous ensemble de $\{1, \dots, n\}$ qui indiquent respectivement les variables sont positives ou négatives. Ainsi la forme générale d'un problème de programmation linéaire est alors

$$\left\{ \begin{array}{ll} \text{minimiser ou maximiser} & \sum_{j=1}^n c_j x_j \\ \text{sous les contraintes} & a_i^T x \geq b_i, \quad i \in M_1, \\ & a_i^T x \leq b_i, \quad i \in M_2, \\ & a_i^T x = b_i, \quad i \in M_3, \\ & x_j \geq 0, \quad j \in N_1, \\ & x_j \leq 0, \quad j \in N_2. \end{array} \right. \quad (1)$$

☞ Les variables $x_1, \dots, x_n$ sont appelée **variables d'optimisation ou de décision**.

☞ Un vecteur x qui vérifie toutes les contraintes est appelé **solution possible ou solution réalisable ou encore solution admissible du problème**.

☞ L'ensemble de toutes les solutions réalisables est appelé **l'ensemble ou la région admissible**

ou l'ensemble des réalisables.

☞ La fonction $c^T x$ est la **fonction objectif** ou **fonction coût**.

☞ Une solution réalisable x^* qui minimise ou maximise la fonction objectif (c'est-à-dire $c^T x^* \leq c^T x$ pour toute solution possible x) est appelée **solution optimale** et la valeur $c^T x^*$ est appelée **valeur optimale** ou **l'optimum** de la fonction objectif.

Enfin il est inutile d'étudier séparément dans ce chapitre les problèmes de maximisation d'un programme linéaire car maximiser une fonction objectif $c^T x$ sous certaines conditions est équivalent à minimiser la fonction objectif $-c^T x$ sous les mêmes conditions. En effet

$$\max f(x) = -\min(-f(x)).$$

où f est une fonction réelle de variable x qui peut être un vecteur. Ainsi, nous allons développer ce chapitre avec les problèmes de minimisation des programmes linéaires.

Tout programme linéaire peut se mettre sous la forme

$$\begin{array}{ll} \text{minimiser} & c^T x \\ \text{sous les contraintes} & Ax \geq b. \end{array} \quad (1.1)$$

où A est une matrice de type (m,n) avec m le nombre de contraintes et n le nombre de variables de décisions.

Les inéquations $Ax \geq b$ seront toujours interprétées par composante c'est-à-dire, pour chaque i , la i -ème composante du vecteur Ax , qui est $a_i^T x$ est supérieure ou égale à la i -ème composante b_i du vecteur b .

Exemple 3. Le problème de programmation linéaire de l'exemple 1 peut se réécrire sous la forme

$$\begin{array}{llllll} \text{minimiser} & 2x_1 - & x_2 + & 4x_3 & & \\ \text{sous les contraintes} & -x_1 - & x_2 - & & x_4 & \geq -2 \\ & & 3x_2 - & x_3 & & \geq 5 \\ & & -3x_2 + & x_3 & & \geq -5 \\ & & & & x_3 + x_4 & \geq 3 \\ & x_1 & & & & \geq 0 \\ & & & & -x_3 & \geq 0 \end{array}$$

qui est de la même forme que (1.1) avec $c^T = (2, -1, 4, 0)$, $x^T = (x_1, x_2, x_3, x_4)$ et $b^T = (-2, 5, -5, 3, 0, 0)$.

$$A = \begin{bmatrix} -1 & -1 & 0 & -1 \\ 0 & 3 & -1 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

1.2.1 Forme canonique

Tout problème de programmation linéaire peut s'écrire sous la forme suivante appelée forme canonique

$$\begin{aligned} & \text{minimiser} && c^T x \\ & \text{sous les contraintes} && Ax \leq b \\ & && x \geq 0, \end{aligned} \tag{1.2}$$

où A est une matrice $m \times n$ (c'est-à-dire m lignes et n colonnes) dont les lignes sont les vecteurs lignes $a_1^T, \dots, a_m^T$ et $b^T = (b_1, \dots, b_m)$.

Les inégalités telles que $Ax \geq b$ sont souvent interprétées par composantes, c'est-à-dire, pour tout i , la i ème composante du vecteur Ax , qui est $a_i^T x$, est supérieure ou égale à la i ème composante b_i du vecteur b .

1.2.2 Forme standard des problèmes

Un problème de programmation linéaire de la forme

$$\begin{aligned} & \text{minimiser} && c^T x \\ & \text{sous les contraintes} && Ax = b \\ & && x \geq 0, \end{aligned} \tag{1.3}$$

est dit sous forme standard.

1.2.3 Réduction à la forme standard d'un problème de programmation linéaire

Toute forme d'un problème de programmation linéaire peut être transformée à la forme standard si elle ne l'est pas et vice versa. Les deux formes dans ce cas sont équivalentes. Dire que les deux formes sont équivalentes signifie que connaissant une solution réalisable de l'une des formes, on peut construire une solution réalisable de l'autre avec la même valeur de la fonction objectif. En particulier les deux formes ont la même valeur optimale et étant donné une solution optimale de l'une, on peut construire une solution optimale de l'autre. La transformation à la forme standard requiert deux étapes :

- (a) Élimination des variables sans contraintes : S'il se trouve dans la forme qu'on veut transformer une variable sans contrainte x_j , alors on le remplace par $x_j^+ - x_j^-$, où x_j^+ et x_j^- sont deux nouvelles variables telles que $x_j^+ \geq 0$ et $x_j^- \geq 0$.
- (b) Élimination des contraintes d'inégalité. Étant donné une contrainte d'inégalité de la forme

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

on introduit une nouvelle variable s_i et la contrainte sous forme standard est donnée par

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j + s_i &= b_i, \\ s_i &\geq 0. \end{aligned}$$

Ces variables s_i s'appellent **variables d'écart**.

Pour les contraintes

$$\sum_{j=1}^n a_{ij}x_j \geq b_i,$$

la forme standard est donnée par

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j - s_i &= b_i, \\ s_i &\geq 0. \end{aligned}$$

Ainsi pour résoudre tout programme linéaire par la méthode algébrique, il suffit de le transformer à la forme standard correspondante puis le résoudre. Pour cela nous allons seulement développer les méthodes permettant de résoudre des problèmes sous standard.

Exemple 4. *Le problème*

$$\begin{aligned} \text{minimiser} \quad & 2x_1 + 4x_2 \\ \text{sous les contraintes} \quad & x_1 + x_2 \geq 3 \\ & 3x_1 + 2x_2 = 14 \\ & x_1 \geq 0, \end{aligned}$$

est équivalent au problème sous forme standard

$$\begin{aligned} \text{minimiser} \quad & 2x_1 + 4x_2^+ - 4x_2^- \\ \text{sous les contraintes} \quad & x_1 + x_2^+ - x_2^- - x_3 = 3 \\ & 3x_1 + 2x_2^+ - 2x_2^- = 14 \\ & x_1, x_2^+, x_2^-, x_3 \geq 0. \end{aligned}$$

Par exemple, étant donné une solution réalisable $(x_1, x_2) = (6, -2)$ du problème original, nous obtenons alors une solution réalisable $(x_1, x_2^+, x_2^-, x_3) = (6, 0, 2, 1)$ du problème sous forme standard, qui a le même coût. Inversement, étant donné une solution réalisable $(x_1, x_2^+, x_2^-, x_3) = (8, 1, 6, 0)$ du problème sous forme standard, nous obtenons une solution réalisable $(x_1, x_2) = (8, -5)$ du problème original avec le même coût.

1.3 Exemples de modélisation de quelques problèmes de PL

Pour formuler un modèle d'optimisation linéaire, il faut :

1. comprendre le problème (souvent difficile dans la pratique)
2. Identifier les variables de décision
3. Formuler les contraintes comme combinaison linéaires des variables de décision
4. Poser la fonction objectif comme combinaison linéaires des variables de décision

Exemple 5 (Problème de fabrication dans une usine). *Une usine produit deux types d'objets : les objets de type A et ceux de type B. L'usine dispose en stock de 1200 kg de matières premières M_1 et de 800kg de matières premières M_2 qui entrent dans la fabrication des objets A et B. La*

réalisation d'un objet de type A nécessite 20kg de M_1 et de 20kg de M_2 . Celle d'un objet B nécessite 30kg de M_1 et 10kg de M_2 . L'usine gagne 400 francs sur la vente d'un objet A et 300 francs sur la vente d'un objet B . Écrivons le programme linéaire qui modélise la maximisation du bénéfice de l'usine sous les contraintes données par le problème lorsque l'usine produit x objets A et y objets B .

Exemple 6 (Problème de transport). Dans deux dépôts D_1 et D_2 sont disponibles respectivement 250 containers et 450 containers. Trois magasins A , B et C ont commandés 200 containers chacun. Les coûts de transport par container (en euro) sont les suivants :

magasin	A	B	C
dépôt D_1	3,4	2,2	2,9
dépôt D_2	3,4	2,4	2,5

Le but du problème est de minimiser le coût total de transport des containers des dépôts vers les magasins en respectant les disponibilités et les demandes.

Pour résoudre ce problème nous allons définir les variables (les inconnues).

Soit x_{ij} le nombre de containers transportés du dépôt i vers le magasin j ($i = 1, 2$ $j = A, B, C$).

Le programme linéaire s'énonce de la façon suivante :

$$\left\{ \begin{array}{l} \text{minimiser} \quad Z(x_{ij}, i = 1, 2 \ j = A, B, C) = 3,4x_{1A} + 2,2x_{1B} + 2,9x_{1C} + 3,4x_{2A} + 2,4x_{2B} + 2,5x_{2C} \\ \text{S.C} \end{array} \right. \left\{ \begin{array}{l} x_{1A} + x_{1B} + x_{1C} \leq 250 \\ x_{2A} + x_{2B} + x_{2C} \leq 450 \end{array} \right\} \text{Disponibilités}$$

$$\left\{ \begin{array}{l} x_{1A} + x_{2A} = 200 \\ x_{1B} + x_{2B} = 200 \\ x_{1C} + x_{2C} = 200 \end{array} \right\} \text{Demandes}$$

$$x_{ij} \geq 0 \text{ à valeurs entières.}$$

Nous pouvons donc généraliser ce problème de transport :

p dépôts $D_1, D_2, \dots, D_p$

q magasins $M_1, M_2, \dots, M_q$

a_i quantité disponible au dépôt D_i , $i = 1, \dots, p$

b_j quantité demandée par le magasin M_j , $j = 1, \dots, q$

f_{ij} coût unitaire de transport de D_i vers M_j

x_{ij} quantité transportée de D_i vers M_j

Programmation linéaire :

$$\left\{ \begin{array}{l} \text{minimiser} \quad Z(x_{ij}, i = 1, \dots, j = 1, \dots, q) = \sum_{i=1}^p \sum_{j=1}^q f_{ij}x_{ij} \\ \text{sous les contraintes} \end{array} \right. \left\{ \begin{array}{l} \sum_{j=1}^q x_{ij} \leq a_i, \quad i = 1, \dots, p \text{ (dépôts)} \\ \sum_{i=1}^p x_{ij} = b_j, \quad j = 1, \dots, q \text{ (magasins)} \\ x_{ij} \geq 0 \text{ (éventuellement à valeurs entières).} \end{array} \right.$$

Remarque 1.3.1. 1. Le choix du signe d'égalité ou d'inégalité dans les contraintes concernant les dépôts et les magasins dépend du contexte concret.

2. $\sum_{i=1}^p a_i \geq \sum_{j=1}^q b_j$ (quantité disponible $\geq$ quantité demandée).

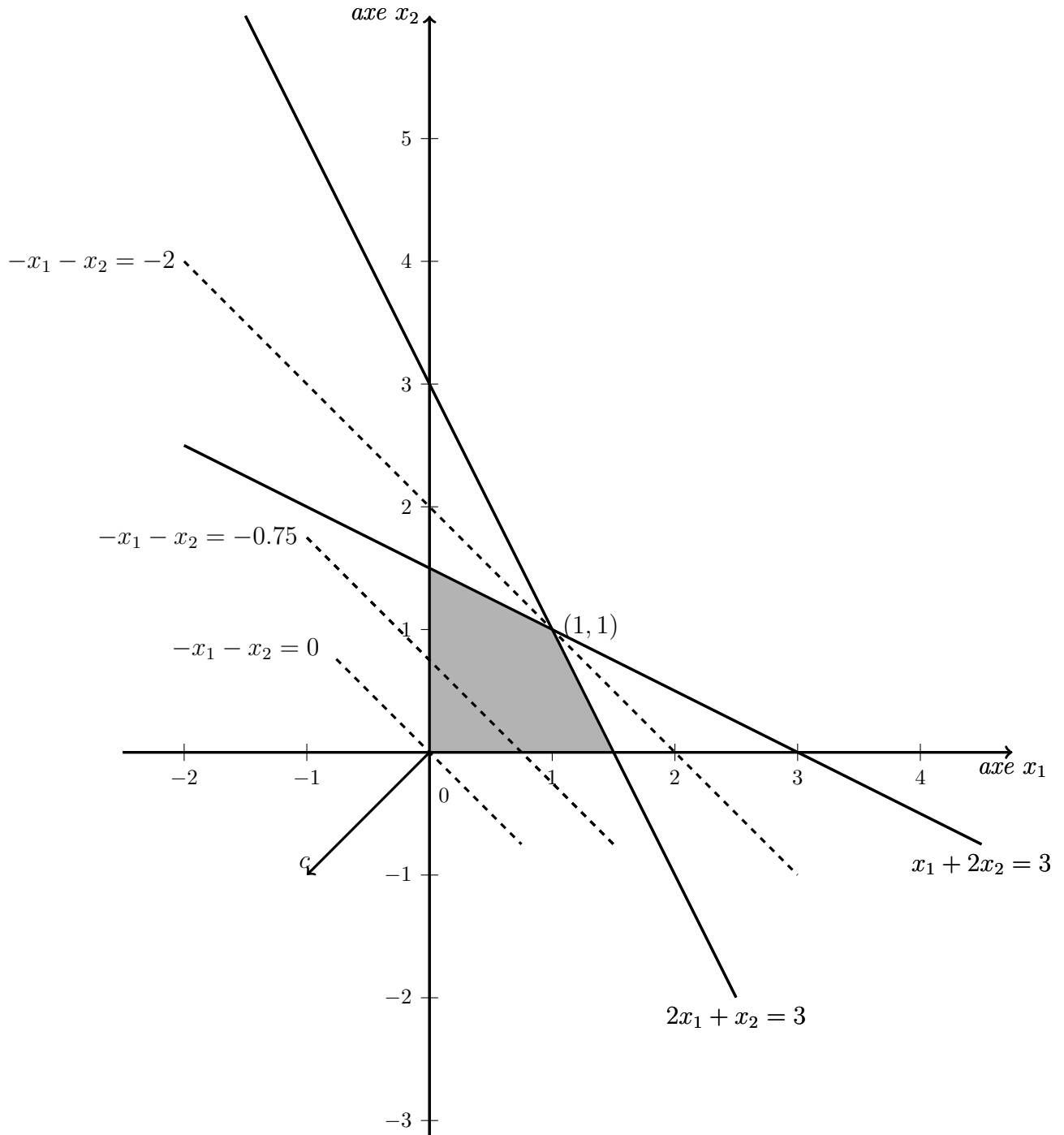
1.4 Résolution par la méthode graphique d'un programme linéaire dépendant de deux ou trois variables

Dans cette section, nous considérons quelques exemples simples qui fournissent des informations géométriques utiles sur la nature des problèmes de programmation linéaire. Considérons l'exemple suivant qui concerne la solution graphique d'un problème de programmation linéaire à deux variables.

Exemple 7.

$$\begin{array}{ll} \text{minimiser} & -x_1 - x_2 \\ \text{sous les contraintes} & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0, \end{array}$$

L'ensemble des solutions possibles est la région ombrée de la figure ci-après. Afin de trouver une solution optimale, nous procédons comme suit. Pour tout réel z donné, nous considérons l'ensemble de tous les points dont le coût $c^T x$ est égal à z ; dans cet exemple c'est la droite décrite par l'équation $-x_1 - x_2 = z$. Notez que cette droite est perpendiculaire au vecteur $c^T = (-1, -1)$. Différentes valeurs de z conduisent à différentes droites, toutes parallèles les unes aux autres. En particulier, augmenter z correspond au déplacement de la droite $z = -x_1 - x_2$ dans la direction du vecteur c . Puisque nous souhaitons minimiser z , nous allons déplacer la droite autant que possible dans la direction du vecteur $-c$, tant que nous ne quittons pas la région des solutions possibles. Dans ce déplacement, le mieux que nous puissions faire est $z = -2$ (voir la figure ci-après), et ainsi, le vecteur $x = (1, 1)$ est une solution optimale. Notons que cette solution optimale est un "coin" ou sommet de l'ensemble des solutions possibles.

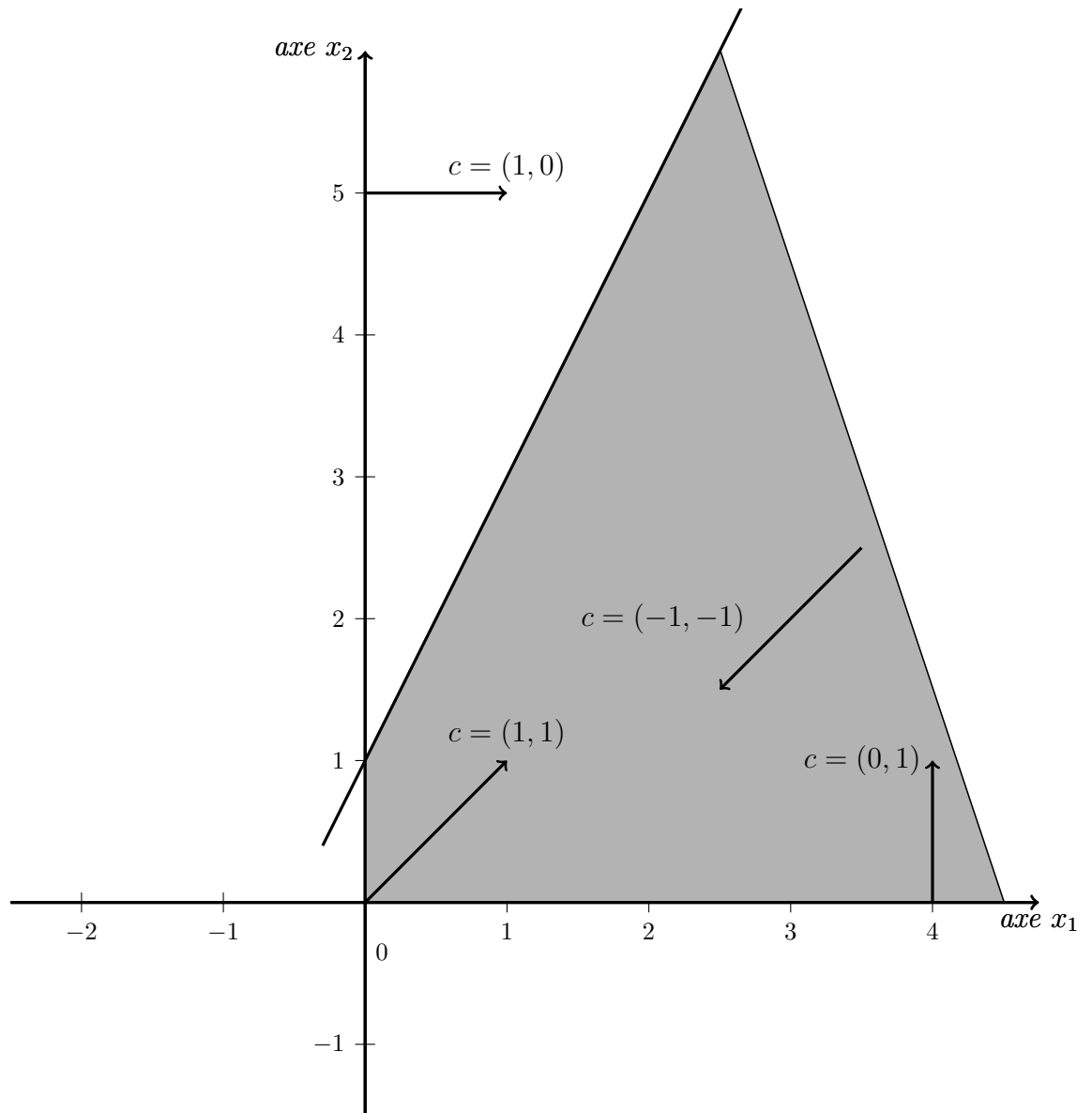


Dans cet exemple l'ensemble des solutions possibles est borné (c-à-d n'est pas étendu à l'infini), et le problème à une solution unique. Cela n'est pas toujours le cas : il existe d'autres possibilités qui sont illustrées dans l'exemple suivant.

Exemple 8. Considérons l'ensemble des solutions réalisables de $\mathbf{R}^2$ défini par les contraintes

$$\begin{cases} -2x_1 + x_2 \leq 1 \\ x_1, x_2 \geq 0. \end{cases}$$

Ici il s'agit de minimiser, sous ces contraintes, une fonction objectif ayant pour vecteur coût c .



1. Pour le vecteur coût $c^T = (1, 1)$, il est clair que $x = (0, 0)$ est l'unique solution optimale.
2. Pour le vecteur coût $c^T = (1, 0)$, il y'a plusieurs solutions optimales, c-à-d tout vecteur x de la forme $x = (0, x_2)$ avec $0 \leq x_2 \leq 1$, est optimal. Notons que l'ensemble des solutions optimales est borné.
3. Pour le vecteur coût $c^T = (0, 1)$, il y'a plusieurs solutions optimales, c-à-d tout vecteur x de la forme $x = (x_1, 0)$ avec $x_1 \geq 0$, est optimal. Dans ce cas l'ensemble des solutions optimales n'est pas borné.
4. Considérons le vecteur coût $c^T = (-1, -1)$. Pour chaque solution possible (x_1, x_2) , nous pouvons toujours trouver une autre solution possible avec moins de coût, en augmentant la valeur de x_1 . Ainsi, aucune solution possible n'est optimale. De plus, en considérant les vecteurs (x_1, x_2) avec des valeurs toujours croissantes de x_1 et x_2 , nous pouvons obtenir une suite de solutions réalisables dont le coût converge vers $-\infty$. On dit donc que le coût optimal est $-\infty$.
5. Si nous imposons une contrainte supplémentaire de la forme $x_1 + x_2 \leq -2$, il est évident qu'aucune solution possible n'existe.

En résumé, selon l'exemple précédent, nous retenons les possibilités suivantes dans la résolution d'un problème de programmation linéaire :

1. il existe une unique solution optimale ;
2. il existe plusieurs solutions optimales ; dans ce cas, l'ensemble des solutions optimales peut être soit borné, soit non borné ;
3. le coût optimal est $-\infty$ et aucune solution possible n'est optimale ;
4. l'ensemble des solutions possibles est vide.

Remarque 1.4.1. *Dans les exemple que nous avons considéré, si le programme linéaire admet au moins une solution optimale, alors une solution optimale peut être trouvée parmi les sommets de l'ensemble des réalisables. Nous démontrerons que ce fait est une caractéristique générale des problèmes de programmation linéaire, à condition que l'ensemble des solutions réalisables ait au moins un sommet.*

1.5 La géométrie de la programmation linéaire

1.5.1 polyèdre et ensembles convexes

Hyperplans, demi-espaces et polyèdre

Définition 1.5.1. *Un polyèdre est un ensemble qui peut être décrit par*

$$\{x \in \mathbb{R}^n / Ax \geq b\},$$

où A est une matrice de type (m, n) et b un vecteur de $\mathbb{R}^m$. Nous savons que l'ensemble des solutions réalisables de tout problème de programmation linéaire peut être décrit par les contraintes d'inégalité de la forme $Ax \geq b$. Ainsi l'ensemble des solutions réalisables d'un programme linéaire est un polyèdre. En particulier un ensemble de la forme

$$\{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$$

est aussi un polyèdre dit polyèdre sous forme standard.

Un polyèdre peut être borné ou s'étendre à l'infini.

Définition 1.5.2. *Un sous ensemble de S de $\mathbb{R}^n$ est borné s'il existe une constante positive C telle que pour tout $x \in S$, $|x| \leq C$ où $|\cdot|$ est la norme euclidienne de $\mathbb{R}^n$.*

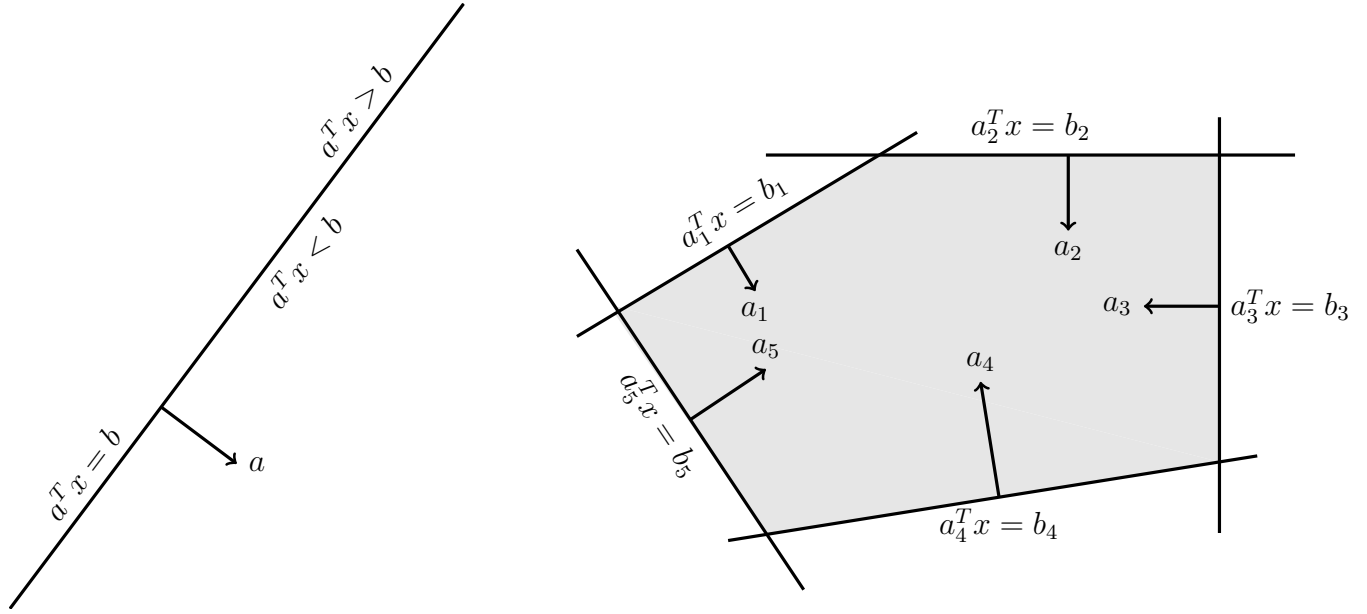
Définition 1.5.3. *Soit $a \in \mathbb{R}^n \setminus \{0\}$ et $b \in \mathbb{R}$.*

a/ L'ensemble $H = \{x \in \mathbb{R}^n / a^T x = b\}$ est appelé hyperplan de $\mathbb{R}^n$.

b/ L'ensemble $D = \{x \in \mathbb{R}^n / a^T x \geq b\}$ est appelé demi-espace de $\mathbb{R}^n$.

Remarque 1.5.1. *l'hyperplan H est la frontière du demi-espace D . De plus, le vecteur a dans la définition de l'hyperplan est perpendiculaire à l'hyperplan lui-même. En effet, pour tout $x, y \in H$, $a^T x = a^T y$. Ainsi $a^T (x - y) = 0$ et par conséquent le vecteur a est orthogonal à tout vecteur de l'hyperplan H .*

Enfin un polyèdre est l'intersection d'un nombre fini de demi-espaces.



La figure de gauche présente un hyperplan et deux demi-espaces ; Celle de droite montre un polyèdre $\{x \mid a_i^T x = b_i, i = 1, \dots, 5\}$ qui est l'intersection de 5 demi-espaces. Notons que chaque vecteur a_i est un vecteur normal de l'hyperplan $\{x \mid a_i^T x = b_i\}$.

Ensembles convexes

Définition 1.5.4. Un ensemble $S \subset \mathbb{R}^n$ est convexe si pour tout $x, y \in S$, pour tout $\lambda \in [0, 1]$, $\lambda x + (1 - \lambda)y \in S$. De façon simple, S est convexe si pour tout $x, y \in S$, le segment formé par x et y appartient à S .

Théorème 1.5.1. (a) Si $\{S_i\}_{i \in I}$ est une famille d'ensembles convexes (I est un ensemble d'indices), alors $\bigcap_{i \in I} S_i$ est un ensemble convexe.

(b) Tout polyèdre est un ensemble convexe.

Démonstration. (a) Soient $x, y \in \bigcap_{i \in I} S_i$ et $\lambda \in [0, 1]$. Pour tout $i \in I$, x et y appartiennent à S_i . Pour tout i , S_i étant convexe, on a $\lambda x + (1 - \lambda)y \in S_i$, pour tout i . Ainsi $\lambda x + (1 - \lambda)y \in \bigcap_{i \in I} S_i$. Alors $\bigcap_{i \in I} S_i$ est convexe.

(b) Soient a un vecteur et b un scalaire. Soient $x, y \in \mathbb{R}^n$ tels que $a^T x \geq b$ et $a^T y \geq b$ c'est-à-dire x et y appartiennent au même demi-espace. Soit $\lambda \in [0, 1]$. Alors

$$a^T(\lambda x + (1 - \lambda)y) = \lambda a^T x + (1 - \lambda)a^T y \geq \lambda b + (1 - \lambda)b = b.$$

Ainsi $\lambda x + (1 - \lambda)y$ appartient au même demi-espace que x et y . Un demi-espace est alors convexe. Un polyèdre étant une intersection finie de demi-espaces, on déduit alors tout polyèdre est convexe d'après (a).

1.6 Bases ou matrices Bases-Solutions de base- Solutions réalisable de base

Considérons un polyèdre $P \subset \mathbb{R}^n$ définie par les contraintes d'égalités et d'inégalités linéaires

$$\begin{aligned} a_i^T x &\geq b_i, & i \in M_1 \\ a_i^T x &\leq b_i, & i \in M_2 \\ a_i^T x &= b_i, & i \in M_3, \end{aligned}$$

où M_1 , M_2 , and M_3 sont des ensemble fini d'indices, chaque a_i est un vecteur de $\mathbb{R}^n$ et chaque b_i est un scalaire.

Définition 1.6.1. Si un vecteur x^* vérifie $a_i^T x^* = b_i$ pour certaines valeurs de i appartenant à M_1 , M_2 ou M_3 , alors on dit que la contrainte correspondante est **active** en x^* .

Théorème 1.6.1. Soit x^* un élément de $\mathbb{R}^n$ et soit $I = \{i | a_i^T x^* = b_i\}$ l'ensemble des indices des contraintes qui sont actives en x^* . Alors, les assertions suivantes sont équivalentes :

- (a) il existe n vecteurs dans l'ensemble $\{a_i | i \in I\}$, qui sont linéairement indépendants.
- (b) L'espace engendré par les vecteurs a_i , $i \in I$, est égal à $\mathbb{R}^n$, c'est-à-dire chaque élément de $\mathbb{R}^n$ peut être exprimé comme une combinaison linéaire des vecteurs a_i , $i \in I$.
- (c) Le système d'équations $a_i^T x = b_i$, $i \in I$, admet une unique solution.

Définition 1.6.2. On considère un polyèdre P défini par des contraintes linéaires exprimées en terme d'équations et d'inéquations. Soit x^* un élément de $\mathbb{R}^n$.

- (a) Le vecteur x^* est une **solution de base** si :
 - (i) toutes les contraintes exprimées en terme d'équation sont actives en x^* ;
 - (ii) Parmi les contraintes qui sont actives en x^* , il en existe n qui sont linéairement indépendantes.
- (b) Si x^* est une solution de base qui vérifie toute les contraintes, alors on dit qu'il est une **solution de base réalisable**.

Exemple 9. Considérons le polyèdre $P = \{(x_1, x_2, x_3) / x_1 + x_2 + x_3 = 1, x_1, x_2, x_3 \geq 0\}$. On donne les vecteurs $u = (0, 0, 1)$, $v = (1, 0, 0)$, $w = (0, 1, 0)$ et $z = (\frac{1}{2}, 0, \frac{1}{2})$.

1. Déterminer les contraintes de P qui sont actives en chacun de ces vecteurs.
2. Déterminer les vecteurs qui sont les solutions de base pour P et ceux qui sont des solutions réalisables de base.

1.7 Polyèdre en forme standard

1.7.1 Définitions

Soit $P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$ un polyèdre sous forme standard avec A une matrice de type (m, n) , b un vecteur de $\mathbb{R}^m$ où m est le nombre de contraintes exprimées sous forme d'équations. On supposons que les m lignes de A sont linéairement indépendantes (cela suppose que $m \leq n$). Le rang de A est alors m c'est-à-dire qu'on peut extraire de A une matrice inversible d'ordre m .

Définition 1.7.1. On appelle **matrice de base** B de A toute matrice carrée d'ordre m extraite de A dont les m colonnes sont m colonnes linéairement indépendantes de A . Il est à remarquer que cette matrice B est inversible car elle a m lignes et m colonnes linéairement indépendantes. Ainsi, B peut s'écrire sous la forme

$$B = \left[\begin{array}{c|c|c|c} & & & \\ A_{B(1)} & A_{B(2)} & \dots & A_{B(m)} \\ & & & \end{array} \right];$$

où $A_{B(1)}, \dots, A_{B(m)}$ sont m colonnes linéairement indépendantes extraites de A . Les colonnes $A_{B(1)}, \dots, A_{B(m)}$ de la matrice B sont appelées **colonnes de base**.

Théorème 1.7.1. On considère les contraintes $Ax = b$ et $x \geq 0$ d'un polyèdre sous forme standard avec A une matrice de type (m, n) dont les lignes sont linéairement indépendantes et b un vecteur

de $\mathbb{R}^m$. Soit $B = \left[\begin{array}{c|c|c|c} & & & \\ A_{B(1)} & A_{B(2)} & \dots & A_{B(m)} \\ & & & \end{array} \right]$ une matrice de base extraite de A . On considère le vecteur $x \in \mathbb{R}^n$ vérifiant $Ax = b$ tel que :

(a) les composantes $x_{B(1)}, \dots, x_{B(m)}$ de x soient déterminés par la relation

$$Bx_B = b \text{ c'est-à-dire } x_B = B^{-1}b. \text{ avec } x_B^T = (x_{B(1)}, \dots, x_{B(m)}).$$

(b) Pour tout $i \neq B(1), \dots, B(m)$, les composantes x_i de x sont toutes nulles : $x_i = 0$. Alors le vecteur x est une solution de base du polyèdre sous forme standard $P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$ appelée **solution de base associée à la matrice de base** B .

Définition 1.7.2. .

1. Le vecteur x_B est appelé **vecteur de base** et ses composantes sont appelées **les variables de base** de la solution de base x .
2. Les composantes x_i de x telles que $x_i = 0$ avec $i \neq B(1), \dots, B(m)$, sont appelées **variables hors base** de la solution de base x associées à la base B .

Remarque 1.7.1. .

1. Les variables de base sont au nombre de m .
2. les variables hors base sont au nombre de $n - m$.

Par rapport à la définition, toutes les solutions de base de (FS) peuvent être construites suivant la procédure suivante.

1.7.2 Procédure pour la construction des solutions de base d'un polyèdre sous forme standard

Les lignes de A étant linéairement indépendantes,

1. Choisir une matrice de base B de A c'est-à-dire choisir m colonnes linéairement indépendantes $A_{B(1)}, \dots, A_{B(m)}$ de A .
2. Pour tout $i \neq B(1), \dots, B(m)$, prendre $x_i = 0$.
3. Résoudre le système des m équations $Ax = b$ pour déterminer les inconnues $x_{B(1)}, \dots, x_{B(m)}$.

Remarque 1.7.2. Une solution de base x d'un polyèdre sous forme standard est une **solution de base réalisable ou solution réalisable de base** si elle est positive c-à-d si tous les variables de base sont positives ($x_{B(1)} \geq 0, \dots, x_{B(m)} \geq 0$).

Exemple 10. Soit le problème de programmation linéaire suivant sous forme standard.

$$(FS) \begin{cases} \text{Min} & -20x_1 & - & 20x_2 & - & 50x_3 & & & & & & & & & \\ \text{s.c} & x_1 & + & x_2 & + & 2x_3 & + & x_4 & & & & & & & = 8, \\ & & & x_2 & + & 6x_3 & & & + & x_5 & & & & & = 12, \\ & x_1 & & & & & & & & & + & x_6 & & & = 4 \\ & & & x_2 & & & & & & & & + & x_7 & = 6 \\ & x_1, & x_2, & x_3, & x_4, & x_5, & x_6, & x_7 & \geq & 0 & & & & . \end{cases}$$

1. Donner la matrice A des contraintes en terme d'égalités. On admet que les lignes de A sont linéairement indépendantes.
2. On considère les matrices de base suivantes $B = [A_1 \ A_2 \ A_3 \ A_7]$, $B' = [A_3 \ A_5 \ A_6 \ A_7]$ et $B'' = [A_4 \ A_5 \ A_6 \ A_7]$ où $A_1, A_2, A_3, A_4, A_5, A_6$ et A_7 sont respectivement la première, la deuxième, la troisième, la quatrième, la cinquième, la sixième et la septième colonne de A .

(a) Vérifier que $C = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix}$ et $D = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ sont respectivement les inverses de B et B' .

- (b) Déterminer les solutions de base associées à chacune de ces matrices de base.
- (c) Ces solutions sont-elles réalisables ?

Résolution

1. La matrice A est donnée par

$$A = \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2. (a) Vérifions que C et D sont respectivement les inverses de B et B' .

$$B = [A_1 \ A_2 \ A_3 \ A_7] = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 6 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{et} \quad B' = [A_3 \ A_5 \ A_6 \ A_7] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pour faire la vérification il suffit de montrer que

$$BC = CB = I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{et} \quad B'D = DB' = I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

je vous laisse faire cela.

(b) $\Leftrightarrow$ Solution de base associée à la base $B = [A_1 \ A_2 \ A_3 \ A_7] = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 6 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$

Soit $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$ la solution de base cherchée. Les variables de base sont alors x_1, x_2, x_3

et x_7 . Posons $x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_7 \end{pmatrix}$.

Les variables hors base sont x_4, x_5, x_6 et on a $x_4 = x_5 = x_6 = 0$.
Déterminons les variables de base. On

$$x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_7 \end{pmatrix} = B^{-1}b = Cb = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix} \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix} = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 6 \end{pmatrix}.$$

Ainsi la solution de base associée à B est $x = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}$.

$\Leftrightarrow$ Solution de base associée à la base $B' = [A_3 \ A_5 \ A_6 \ A_7] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Soit $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$ la solution de base cherchée. Les variables de base sont alors x_3, x_5, x_6

et x_7 . Posons $x_B = \begin{pmatrix} x_3 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$.

Les variables hors base sont x_1, x_2, x_4 et on a $x_1 = x_2 = x_4 = 0$.

Déterminons les variables de base. On a

$$x_B = \begin{pmatrix} x_3 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = B'^{-1}b = Db = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix} = \begin{pmatrix} 4 \\ -12 \\ 4 \\ 6 \end{pmatrix}.$$

Ainsi la solution de base associée à B' est $x = \begin{pmatrix} 0 \\ 0 \\ 4 \\ 0 \\ -12 \\ 4 \\ 6 \end{pmatrix}$.

☞ Solution de base associée à la base $B'' = [A_4 \ A_5 \ A_6 \ A_7] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Soit $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$ la solution de base cherchée. Les variables de base sont alors x_4, x_5, x_6

et x_7 . Posons $x_B = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$.

Les variables hors base sont x_1, x_2, x_3 et on a $x_1 = x_2 = x_3 = 0$.

Déterminons les variables de base. On a

$$x_B = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = B''^{-1}b = B''b = b = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}.$$

Ainsi la solution de base associée à B'' est $x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 8 \\ 12 \\ 4 \\ 6 \end{pmatrix}$.

(c) ✎ La solution de base associée à B est $x = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}$. Cette solution de base est une

solution réalisable car toutes les variables de décision sont positives. ✎ La solution de

base associée à B' est $x = \begin{pmatrix} 0 \\ 0 \\ 4 \\ 0 \\ -12 \\ 4 \\ 6 \end{pmatrix}$.

Cette solution de base n'est pas réalisable car $x_5 = -12 < 0$. ✎ La solution de base

associée à B'' est $x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 8 \\ 12 \\ 4 \\ 6 \end{pmatrix}$. Cette solution est réalisable car toutes les variables sont

positives.

1.7.3 Cas où le rang de la matrice A est inférieur au nombre de lignes de A .

Théorème 1.7.2. Soit

$$(P) \begin{cases} Ax = b \\ x \geq 0, \end{cases}$$

un polyèdre sous forme standard où A est une matrice de type (m, n) ($m < n$) avec $a_1^T, \dots, a_m^T$ ses lignes. Supposons que $\text{rang}(A) = k < m$ et que les lignes $a_{i_1}^T, \dots, a_{i_k}^T$ de A sont linéairement

indépendantes. Considérons le polyèdre

$$(Q) \begin{cases} a_{i_1}^T x = b_{i_1} \\ \vdots \\ a_{i_k}^T x = b_{i_k} \\ x \geq 0. \end{cases}$$

Alors $(Q) = (P)$.

Exemple 11. Considérons le polyèdre non vide défini par les contraintes

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 2 \\ x_1 + x_2 &= 1 \\ x_1 + x_3 &= 1 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

La matrice A correspondante a pour rang 2. En effet les deux dernières lignes de cette matrice sont linéairement indépendantes, mais la première ligne est la somme des deux autres. Ainsi la première contrainte est redondante. Après élimination de cette contrainte, le programme qu'on obtient est le même que le programme initial.

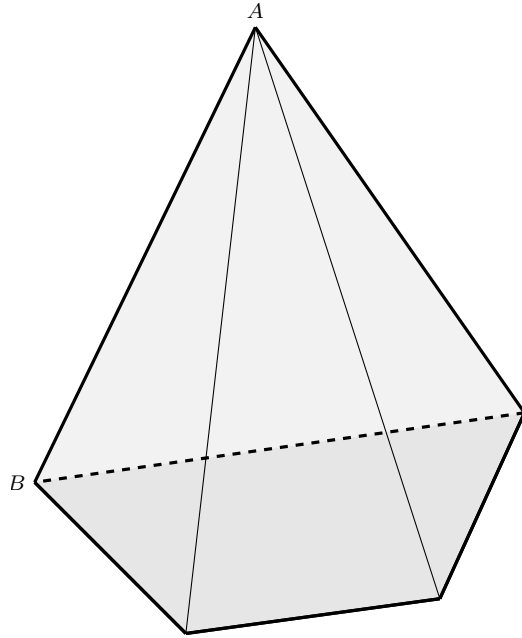
1.8 Dégénérescence

Définition 1.8.1. Une solution de base $x \in \mathbb{R}^n$ est dite dégénérée s'il y'a plus de n contraintes qui sont actives en x .

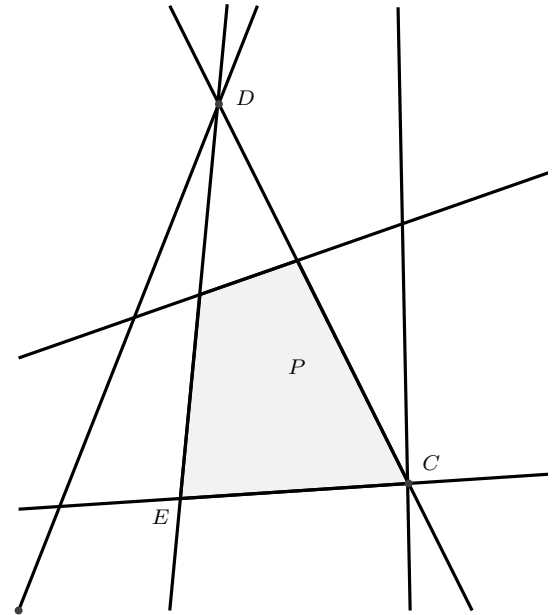
Exemple 12. Considérons le polyèdre P défini par les contraintes

$$\begin{aligned} x_1 + x_2 + 2x_3 &\leq 8 \\ x_2 + 6x_3 &\leq 12 \\ x_1 &\leq 4 \\ x_2 &\leq 6 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Le vecteur $x = (2, 6, 0)$ est une solution de base réalisable non dégénérée car il existe exactement trois contraintes actives et linéairement indépendantes telles que $x_1 + x_2 + 2x_3 \leq 8$, $x_2 \leq 6$ et $x_3 \geq 0$. Le vecteur $(4, 0, 2)$ est une solution de base réalisable dégénérée car il existe exactement trois contraintes actives et linéairement indépendantes telles que $x_1 + x_2 + 2x_3 \leq 8$, $x_2 + 6x_3 \leq 12$, $x_1 \leq 4$ et $x_2 \geq 0$.



(a)



(b)

Les point A et C sont des solutions réalisables de base dégénérées. Les points B et E sont des solutions réalisables de base non dégénérées. Le point D est une solution est solution de base dégénérée.

Dégénérescence d'un programme linéaire en forme standard

En une solution de base, les m contraintes exprimées en terme d'égalité d'un polyèdre en forme standard et les $n - m$ contraintes de la forme $x_i \geq 0$ (avec i parcourant les indices des variables hors base) sont actives en une solution de base donnée. Ainsi avoir plus de n contraintes actives revient au même à dire avoir plus de $n - m$ variables nulles. Ce qui conduit à la définition suivante qui est un cas spécial de la définition 1.8.1.

Définition 1.8.2. Soit

$$(P) \left\{ \begin{array}{ll} \text{minimiser} & c^T x \\ \text{sous les contraintes} & Ax = b \\ & x \geq 0, \end{array} \right.$$

soit x^* une solution de base du polyèdre (P) . Soit m le nombre de lignes de la matrice A . La solution de base x^* est **solution de base dégénérée** de (P) si plus de $n - m$ ($m < n$) composantes de x^* sont nulles c'est-à-dire s'il existe au moins une variable de base de x^* qui est nulle.

Exemple 13. En introduisant les variables d'écart $x_4, \dots, x_7$ dans les contraintes du polyèdre de l'exemple (12) nous pouvons le transformer en forme standard $P = \{x \in \mathbb{R}^7 | Ax = b, x \geq 0\}$, où

$$A = \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}.$$

Considérons la matrice base B constituée des colonnes linéairement indépendantes $A_1, A_2, A_3,$

$$A_7 : B = [A_1 \ A_2 \ A_3 \ A_7] = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 6 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}. \text{ On a}$$

$$B^{-1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix}.$$

Soit $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$ la solution de base associée à B . Les variables de base sont alors x_1, x_2, x_3 et x_7 .

Posons $x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_7 \end{pmatrix}$ le vecteur des variables de base.

Les variables hors base sont x_4, x_5, x_6 et on a $x_4 = x_5 = x_6 = 0$.

Déterminons les variables de base. On a

$$x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_7 \end{pmatrix} = B^{-1}b = Cb = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix} \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix} = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 6 \end{pmatrix}.$$

Ainsi la solution de base associée à B est $x = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}$.

C'est une solution réalisable dégénérée car il y'a plus de $n - m = 7 - 4 = 3$ zéro c'est-à-dire qu'il y'a au moins une variable de base qui est nulle (la variable de base x_2 est nulle).

Exercice 1.1. Les vecteurs $(1, 1, 0)$ et $(0, 0, 1)$ sont-ils dégénéré ou non pour le système de contraintes

$$\begin{array}{rclcl} x_1 & - & x_2 & & = & 0 \\ x_1 & + & x_2 & + & 2x_3 & = & 2 \\ & & x_1, & x_2, & x_3 & \geq & 0 \end{array} \quad ?$$

1.9 Existence des points extrêmes

Définition 1.9.1. Un polyèdre $P \subset \mathbb{R}^n$ contient une droite ou une ligne s'il existe un vecteur $x \in P$ et un vecteur non nul $d \in \mathbb{R}^n$ tels que $x + \lambda d \in P$ pour tous scalaire λ .

Théorème 1.9.1. Supposons que le polyèdre $P = \{x \in \mathbb{R}^n / a_i^T x \geq b_i, i = 1, \dots, m\}$ est non vide avec $m \geq n$. Alors, les assertions suivantes sont équivalentes :

- (a) le polyèdre P admet au moins un point extrême ;
- (b) le polyèdre P ne contient pas de ligne
- (c) il existe n vecteurs linéairement indépendants qu'on peut extraire de la famille $a_1, \dots, a_m$.

Corollaire 1.9.1. Tout polyèdre borné non vide et tout polyèdre non vide sous forme standard possède au moins une solution de base réalisable.

1.10 Optimalité des points extrêmes

Théorème 1.10.1. Considérons le problème de programmation linéaire consistant à minimiser $c^T x$ sur un polyèdre P . Supposons que P possède au moins un point extrême et qu'il existe une solution optimale. Il existe alors une solution optimale qui est un point extrême de P .

Théorème 1.10.2. Considérons le problème de programmation linéaire consistant à minimiser $c^T x$ sur un polyèdre P . Supposons que P possède au moins un point extrême. Alors, soit le coût optimal est égal à $-\infty$, soit il existe un point extrême qui est optimal.

Corollaire 1.10.1. Considérons le problème de programmation linéaire consistant à minimiser $c^T x$ sur un polyèdre non vide. Alors, soit le coût optimal est égal à $-\infty$, soit il existe une solution optimale.

1.11 La méthode du Simplexe

Nous avons vu au chapitre 1 que si un problème de programmation linéaire sous forme standard admet une solution optimale, alors il existe une solution réalisable de base qui est optimale. La méthode du simplexe est basée sur ce fait et recherche une solution optimale en passant d'une solution réalisable de base à une autre, le long des arêtes de l'ensemble des solutions possibles, toujours dans une direction de réduction des coûts. Finalement, une solution réalisable de base optimale est celle pour laquelle aucune des arêtes disponibles ne conduit à une réduction des coûts ; dans ce cas l'algorithme se termine.

Dans ce chapitre, nous fournissons un développement détaillé de la méthode du simplexe et discutons de quelques implémentations différentes, notamment le tableau du simplexe et la méthode du simplexe révisée. Nous abordons également certaines difficultés qui peuvent survenir en présence de dégénérescence.

Dans tout le chapitre, nous considérons le problème sous forme standard

$$\begin{array}{ll} \text{minimiser} & c^T x \\ \text{sous les contraintes} & Ax = b \\ & x \geq 0, \end{array}$$

et nous posons P le polyèdre constitué de l'ensemble des réalisables correspondant. Nous supposons que A est une matrice de dimensions $m \times n$ et que ses lignes sont linéairement indépendantes. Dans toute la suite A_i est la i ème colonne de A et a_i^T ou a'_i est la i ème ligne de A .

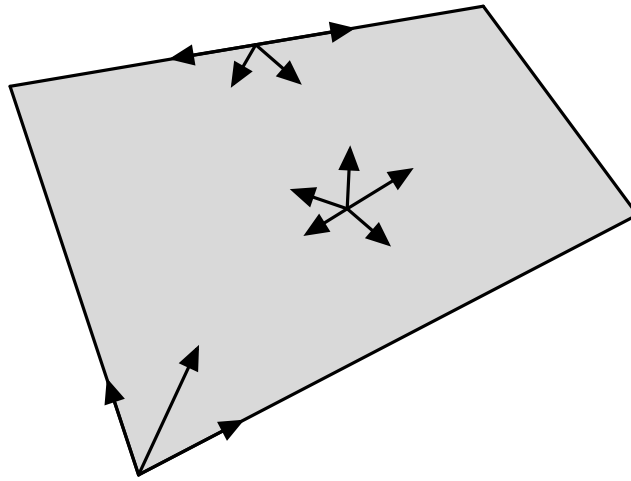
1.11.1 Conditions d'optimalité

De nombreux algorithmes d'optimisation sont structurés comme suit : étant donné une solution réalisable, on recherche dans son voisinage une solution réalisable qui lui est proche avec un coût plus petit. Si aucune solution réalisable proche ne conduit à une amélioration du coût, l'algorithme se termine et on obtient alors une solution localement optimale. De façon général, une solution localement optimale d'un problème d'optimisation n'est pas forcément une solution globalement optimale. Mais, en programmation linéaire une solution optimale locale est une solution optimale globale : cela vient du fait qu'on minimise une fonction convexe (la fonction objectif linéaire) sur un ensemble convexe qui est le polyèdre.

Objectifs de cette section :

Étant donné une solution réalisable x d'un problème sous forme standard, nous nous concentrons sur la recherche d'une direction conduisant vers une autre solution réalisable proche de x dont le coût est moins que celui de x . Nous allons ainsi donner les conditions d'optimalité associées. Les conditions d'optimalité sont les conditions pour lesquelles on reconnait qu'une solution réalisable de base est optimale pour un problème de programmation linéaire sous forme standard.

Supposons qu'on se trouve au point $x \in P$ et on décide de s'éloigner de x dans la direction d'un vecteur d de $\mathbb{R}^n$. De façon clair, on choisira seulement les directions d qui ne conduisent pas immédiatement hors de l'ensemble des solutions réalisables (polyèdre) ; ce qui conduit à la définition suivante :



Définition 1.11.1. Soit x un élément du polyèdre P . Un vecteur d de $\mathbb{R}^n$ est dit *direction réalisable* au point x , s'il existe un réel positif θ pour lequel $x + \theta d \in P$.

Soit x une solution de base réalisable d'un problème sous forme standard ; soient $B_{(1)}, \dots, B_{(m)}$ les indices des variables de base et $B = [A_{B(1)}, \dots, A_{B(m)}]$ la matrice de base correspondante. En particulier, nous avons $x_i = 0$ pour chaque variable non basique, tandis que le vecteur $x_B^T = (x_{B(1)}, \dots, x_{B(m)})$ des variables de base est donné par

$$x_B = B^{-1}b.$$

Nous considérons la possibilité de s'éloigner de x , vers un nouveau vecteur $x + \theta d$, en sélectionnant une variable hors base x_j (qui est initialement égal zéro), et en l'augmentant à une valeur $\theta > 0$, tout en gardant les autres variables hors base restantes égal à zéro. Algébriquement, $d_j = 1$ et $d_i = 0$ pour tout indice i de variable hors base autre que j . Dans ce cas, le vecteur x_B des variables de base devient $x_B + \theta d_B$, où $d_B^T = (d_B(1), d_B(2), \dots, d_B(m))$ est le vecteur dont les composantes sont ceux de d ayant pour indices, ceux des variables de base.

Étant donné que nous ne nous intéressons qu'aux solutions réalisables, nous avons

$$A(x + \theta d) = b \quad (1.4)$$

$$x + \theta d \geq 0 \quad (1.5)$$

Comme x est réalisable, nous avons $Ax = b$. Ainsi, pour $\theta > 0$, (1.4) entraîne que $Ad = 0$. Rappelons maintenant que $d_j = 1$ et que $d_i = 0$ pour tous les autres indices des autres variables hors base, alors

$$0 = Ad = \sum_{i=1}^n A_i d_i = \sum_{i=1}^m A_{B(i)} d_{B(i)} + A_j = B d_B + A_j.$$

Puisque la matrice de base B est inversible, on a

$$d_B = B^{-1} A_j. \quad (1.6)$$

Le vecteur de direction d que nous venons de construire sera appelé j -ième direction de base. Nous avons jusqu'ici garanti que les contraintes d'égalité (1.4) sont respectées lorsque l'on s'éloigne de x le long de la direction de base d . Qu'en est-il des contraintes de positivité (1.5)? Nous rappelons que la variable x_j est augmentée de $\theta > 0$, et que toutes les autres variables hors base restent à zéro. Ainsi, nous n'avons à nous soucier que des variables de base. Nous distinguons deux cas :

- (a) Supposons que x soit une solution de base réalisable non dégénérée. Alors $x_B > 0$. d'où il résulte que $x_B + \theta d_B \geq 0$, et la faisabilité est maintenue, lorsque θ est suffisamment petit. En particulier, d est une direction réalisable.
- (b) Supposons maintenant que x soit dégénéré. Alors, d n'est pas toujours une direction réalisable. En effet, si la variable de base $x_{B(i)}$ est nulle, tandis que la composante correspondante $d_{B(i)}$ de $d_B = -B^{-1} A_j$ est strictement négative, alors $x_{B(i)} + \theta d_{B(i)} < 0$. Ainsi en suivant la j -ième direction de base, la contrainte de positivité pour $x_{B(i)} + \theta d_{B(i)}$ est immédiatement violée, et nous sommes conduits à des solutions irréalisables.

Nous étudions maintenant les effets sur la fonction de coût si nous nous déplaçons le long d'une direction de base. Si d est la j -ième direction de base, alors le taux $c^T d$ de variation de coût le long de la direction d est donné par $c_B^T d_B + c_j$, où $c_B^T = (c_{B(1)}, \dots, c_{B(m)})$. En utilisant (1.6), cela revient au même que $c_j - c_B^T B^{-1} A_j$. Cette quantité est suffisamment importante pour justifier une définition.

Définition 1.11.2 (coût réduit). .

Soit x une solution de base, soit $B = [A_{B(1)} \dots A_{B(m)}]$ une matrice de base associée, et soit $c_B^T = (c_{B(1)}, \dots, c_{B(m)})$ le vecteur des coûts des variables de base. Pour chaque j , nous définissons le **coût réduit** $\bar{c}_j$ des variables x_j par la formule

$$\bar{c}_j = c_j - c_B^T B^{-1} A_j$$

Remarque 1.11.1. Le coût réduit d'une variable de base est toujours nul c'est-à-dire

$$\bar{c}_{B(i)} = 0, \forall i = B(1), \dots, B(m).$$

Théorème 1.11.1. *Considérons une solution de base réalisable x dont la base qui est associée est B et $\bar{c}$ le vecteur des coûts réduits correspondant.*

- (a) *Si $\bar{c} \geq 0$, alors x est optimale*
- (b) *Si x est optimale et non dégénérée, alors $\bar{c} \geq 0$.*

Remarque 1.11.2. .

- (i) *D'après le (a) du théorème 1.11.1 si x n'est pas optimale, alors il existe $\bar{c}_j < 0$ où j est l'indice d'une certaine variable hors base.*
- (ii) *D'après le (b) du théorème 1.11.1, il est possible que x soit une solution dégénérée optimale, mais que $\bar{c}_j < 0$ où $\bar{c}_j$ est le coût réduit d'une certaine variable hors base.*
- (iii) *Selon le théorème 1.11.1, pour vérifier qu'une solution de base réalisable non dégénérée est optimale, il suffit juste de vérifier si tous les coûts réduits sont positifs.*

Définition 1.11.3. *Une matrice de base B est dit optimale si*

- (a) $B^{-1}b \geq 0$, et
- (b) $\bar{c}^T = c^T - c_B^T B^{-1}A \geq 0^T$.

De façon clair, si une base optimale est trouvée, alors la solution de base correspondante est réalisable, satisfait aux conditions d'optimalité et par conséquent est optimale.

Exemple 14. *Soit le problème de programmation linéaire suivant sous forme standard.*

$$(FS) \left\{ \begin{array}{llllllll} \text{Min} & -20x_1 & - & 20x_2 & - & 50x_3 & & & & & \\ \text{s.c} & x_1 & + & x_2 & + & 2x_3 & + & x_4 & & & = 8, \\ & & & x_2 & + & 6x_3 & & + & x_5 & & = 12, \\ & x_1 & & & & & & & + & x_6 & = 4 \\ & & & x_2 & & & & & & + & x_7 = 6 \\ & x_1, & x_2, & x_3, & x_4, & x_5, & x_6, & x_7 & \geq & 0 & . \end{array} \right.$$

1. *Donner la matrice A des contraintes en terme d'égalités. On admet que les lignes de A sont linéairement indépendantes.*
2. *On considère les matrices de base suivantes $B = [A_1 \ A_2 \ A_3 \ A_7]$, $B' = [A_3 \ A_5 \ A_6 \ A_7]$ et $B'' = [A_4 \ A_5 \ A_6 \ A_7]$ où $A_1, A_2, A_3, A_4, A_5, A_6$ et A_7 sont respectivement la première, la deuxième, la troisième, la quatrième, la cinquième, la sixième et la septième colonne de A .*

(a) Vérifier que $C = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix}$ et $D = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ sont respectivement

les inverses de B et B' .

- (b) *Déterminer les solutions de base associées à chacune de ces matrices de base.*
- (c) *Calculer les coûts réduits des variables $x_1, x_2, x_3, x_4, x_5, x_6, x_7$ pour chaque solution réalisable de base trouvée à la question précédente.*
- (d) *Déduire si possible la solution optimale du problème sous forme standard. Préciser la base optimale de ce programme. Déterminer la valeur optimale de la fonction objectif.*

Résolution

1. La matrice A est donnée par

$$A = \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2. (a) Vérifions que C et D sont respectivement les inverses de B et B' .

$$B = [A_1 \ A_2 \ A_3 \ A_7] = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 6 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{et} \quad B' = [A_3 \ A_5 \ A_6 \ A_7] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pour faire la vérification il suffit de montrer que

$$BC = CB = I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{et} \quad B'D = DB' = I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

je vous laisse faire cela.

$$(b) \Leftrightarrow \text{Solution de base associée à la base } B = [A_1 \ A_2 \ A_3 \ A_7] = \begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 6 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Soit $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$ la solution de base cherchée. Les variables de base sont alors x_1, x_2, x_3

et x_7 . Posons $x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_7 \end{pmatrix}$.

Les variables hors base sont x_4, x_5, x_6 et on a $x_4 = x_5 = x_6 = 0$.

Déterminons les variables de base. On

$$x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_7 \end{pmatrix} = B^{-1}b = Cb = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix} \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix} = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 6 \end{pmatrix}.$$

Ainsi la solution de base associée à B est $x = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}$.

☞ Solution de base associée à la base $B' = [A_3 \ A_5 \ A_6 \ A_7] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Soit $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$ la solution de base cherchée. Les variables de base sont alors x_3, x_5, x_6

et x_7 . Posons $x_B = \begin{pmatrix} x_3 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$.

Les variables hors base sont x_1, x_2, x_4 et on a $x_1 = x_2 = x_4 = 0$.
Déterminons les variables de base. On a

$$x_B = \begin{pmatrix} x_3 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = B'^{-1}b = Db = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix} = \begin{pmatrix} 4 \\ -12 \\ 4 \\ 6 \end{pmatrix}.$$

Ainsi la solution de base associée à B' est $x = \begin{pmatrix} 0 \\ 0 \\ 4 \\ 0 \\ -12 \\ 4 \\ 6 \end{pmatrix}$.

☞ Solution de base associée à la base $B'' = [A_4 \ A_5 \ A_6 \ A_7] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Soit $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$ la solution de base cherchée. Les variables de base sont alors x_4, x_5, x_6

et x_7 . Posons $x_B = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$.

Les variables hors base sont x_1, x_2, x_3 et on a $x_1 = x_2 = x_3 = 0$.
Déterminons les variables de base. On a

$$x_B = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = B''^{-1}b = B''b = b = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}.$$

Ainsi la solution de base associée à B'' est $x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 8 \\ 12 \\ 4 \\ 6 \end{pmatrix}$.

(c) Coûts réduits des variables des solutions réalisables de base trouvées à la question précé-

dente. ☞ Le vecteur $x = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}$ est une solution réalisable de base associée à la base B .

Nous savons que pour cette solution, les variables de base sont x_1, x_2, x_3 et x_7 . Donc les coûts réduits de ces variables sont nuls : $\bar{c}_1 = \bar{c}_2 = \bar{c}_3 = \bar{c}_7 = 0$.

Calculons les coûts réduits des variables hors base x_4, x_5 et x_6 de cette solution réalisable de base.

$$c_B^T = (-20 \ -20 \ -50 \ 0) \text{ et } A = \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned}\bar{c}_4 &= c_4 - c_B^T B^{-1} A_4 = 0 - (-20 \ -20 \ -50 \ 0) \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{35}{2} \\ \bar{c}_5 &= c_5 - c_B^T B^{-1} A_5 = 0 - (-20 \ -20 \ -50 \ 0) \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{5}{2} \\ \bar{c}_6 &= c_6 - c_B^T B^{-1} A_6 = 0 - (-20 \ -20 \ -50 \ 0) \begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{3}{2} & \frac{-1}{2} & \frac{-3}{2} & 0 \\ \frac{-1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{-3}{2} & \frac{1}{2} & \frac{3}{2} & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \frac{5}{2}.\end{aligned}$$

☞ Le vecteur $x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 8 \\ 12 \\ 4 \\ 6 \end{pmatrix}$ est une solution réalisable de base associée à la base B'' . Les

variables de base de cette solution sont x_4, x_5, x_6 et x_7 . Donc le coût réduit de chacune de ces variables de base est nul : $\bar{c}_4 = \bar{c}_5 = \bar{c}_6 = \bar{c}_7 = 0$. Calculons les coûts réduits des variables hors base x_1, x_2 et x_3 de cette solution réalisable de base.

$$c_B^T = (0 \ 0 \ 0 \ 0)$$

$$A = \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned}\bar{c}_1 &= c_1 - c_{B''}^T B''^{-1} A_1 = c_1 = -20 \\ \bar{c}_2 &= c_2 - c_{B''}^T B''^{-1} A_2 = c_2 = -20 \\ \bar{c}_3 &= c_3 - c_{B''}^T B''^{-1} A_3 = c_3 = -50.\end{aligned}$$

(d) D'après le théorème 1.11.1 du cours, la solution réalisable de base $x = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 6 \end{pmatrix}$ associée à la base B est optimale car tous les coûts réduits des variables de cette solution sont positifs.
La matrice de base B est alors la base optimale.

Valeur optimale de la fonction objectif : $v_{opt} = -20 \times 4 - 20 \times 0 - 50 \times 2 = -180$

1.11.2 Développement de l'algorithme du simplexe

Nous allons maintenant terminer le développement de la méthode du simplexe. Notre tâche principale est de déterminer les détails de la manière de passer à une meilleure solution réalisable de base, chaque fois qu'une direction de base rentable est découverte. **Supposons que chaque solution réalisable de base soit non dégénérée.** Cette hypothèse restera en vigueur jusqu'à avis contraire dans cette section. Supposons que nous soyons à une solution réalisable de base x et que nous ayons calculé les coûts réduits $\bar{c}_j$ des variables hors base. Si tous sont positifs, le théorème 1.11.1 montre que nous avons une solution optimale, et nous nous arrêtons. Si par contre, le coût réduit $\bar{c}_j$ d'une variable hors base est négatif, la j -ième direction de base d est une direction réalisable de diminution des coûts. (on rappelle que c'est la direction obtenue en posant $d_j = 1$, $d_i = 0$ pour $i \neq B(1), \dots, B(m)$, j , et $d_B = -B^{-1}A_j$.) En se déplaçant dans cette direction d , la variable hors base x_j devient strictement positive et toutes les autres variables hors base demeurent égaux à zéro. On décrit cette situation en disant que x_j (ou A_j) **entre dans la base**.

Une fois qu'on commence à se déplacer à partir de x le long de la direction d , on parcourt alors les points de la forme $x + \theta d \in P$, où $\theta \geq 0$. Puisque le coût décroît le long de la j ème direction d , il est souhaitable de se déplacer le plus loin possible pour obtenir le plus petit coût le long de cette direction. Cela conduit au point $x + \theta^* d$, où

$$\theta^* = \max \{ \theta \geq 0 \mid x + \theta d \in P \}.$$

Ainsi la valeur qui réduit le coût est alors $\theta^* c^T d$ qui est encore égal $\theta^* \bar{c}_j$.

Déterminons maintenant une formule pour le calcul de θ^* .

Étant donné que $Ad = 0$, on a

$$A(x + \theta d) = Ax = b, \forall \theta.$$

Ce qui signifie que pour tout θ , le vecteur $x + \theta d$ vérifie toutes les contraintes exprimées en terme d'égalité dans le polyèdre P . Dans ce cas, $x + \theta d$ serait non réalisable seulement si l'une de ses composantes est strictement négative. On distingue deux cas :

- (a) si $d \geq 0$, alors $x + \theta d \geq 0$ pour tout $\theta \geq 0$. Ce qui signifie que le vecteur $x + \theta d$ ne peut jamais être irréalisable car pour tout $\theta \geq 0$ va toujours vérifier toutes les contraintes de P . En conséquence, $\theta^* = +\infty$.
- (b) Si d n'est pas positif, c'est-à-dire si $d_i < 0$ pour un certain i , alors la contrainte $x_i + \theta d_i \geq 0$ devient $\theta \leq -\frac{x_i}{d_i}$. Cette contrainte sur θ doit être satisfaite pour tout indice i avec $d_i < 0$.

Ainsi, la plus grande valeur possible de θ pour lequel $x + \theta d \in P$ est

$$\theta^* = \min_{\{i|d_i < 0\}} \left(-\frac{x_i}{d_i} \right) \quad (1.7)$$

Rappelons que si x_i est une variable hors base, alors soit x_i est la variable entrante et $d_i = 1$ ou sinon $d_i = 0$. En d'autres termes, pour tout indice i d'une variable hors base, $d_i \geq 0$. Ainsi, dans (1.7), on a juste besoin de considérer que les variables de base et finalement la formule (1.7) devient

$$\theta^* = \min_{\{i=1,\dots,m|d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right). \quad (1.8)$$

Il faut noter que $\theta^* > 0$ car $x_{B(i)} > 0$ puisqu'on a supposé que les solutions réalisables de base sont non dégénérées.

Exemple 15. On considère le programme linéaire

$$\begin{array}{llllll} \text{minimiser} & c_1x_1 & + & c_2x_2 & + & c_3x_3 & + & c_4x_4 \\ \text{S.C} & x_1 & + & x_2 & + & x_3 & + & x_4 & = & 2 \\ & 2x_1 & & & + & 3x_3 & + & 4x_4 & = & 2 \\ & x_1, & x_2, & x_3, & x_4 & \geq & 0 & & & \end{array}$$

Les deux premières colonnes de la matrice A sont $A_1 = (1, 2)$ et $A_2 = (1, 0)$. Puisqu'elles sont linéairement indépendantes, on peut choisir x_1 et x_2 comme les variables de base. La matrice de base correspondante est

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

Ainsi on obtient une solution réalisable de base associée à B est donnée par $x^T = (1 \ 1 \ 0 \ 0)$. Cette solution est non dégénérée.

Une fois θ^* est choisi, et est supposé fini, on peut se déplacer de la solution réalisable de base x vers la solution réalisable de base $y = x + \theta^*d$. Comme $x_j = 0$ et $d_j = 1$, on a $y_j = \theta^* > 0$. Soit ℓ un indice qui minimise (1.8), c'est-à-dire

$$-\frac{x_{B(\ell)}}{d_{B(\ell)}} = \min_{\{i=1,\dots,m|d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right) = \theta^*. \quad (1.9)$$

On a

$$x_{B(\ell)} + \theta^*d_{B(\ell)} = 0.$$

On observe alors que la variable de base $x_{B(\ell)}$ devient zéro c'est-à-dire que $y_{B(\ell)} = 0$ cependant la variable hors base x_j est devenue $y_j = \theta^* > 0$. Ce qui suggère que x_j remplace $x_{B(\ell)}$ dans la base. Dans le même sens, on prend l'ancienne matrice de base B et on remplace la colonne $A_{B(\ell)}$ par la colonne A_j et on obtient ainsi une nouvelle matrice

$$\overline{B} = \left[\begin{array}{c|c|c|c|c|c|c} | & & | & | & | & & | \\ A_{B(1)} & \dots & A_{B(\ell-1)} & A_{B(j)} & A_{B(\ell+1)} & \dots & A_{B(m)} \\ | & & | & | & | & & | \end{array} \right]. \quad (1.10)$$

De façon équivalente, cela signifie qu'on a remplacé l'ensemble des indices de base $B(1), \dots, B(m)$ par un nouveau ensemble d'indices donné par

$$\overline{B}(i) = \begin{cases} B(i), & \text{si } i \neq \ell \\ j, & \text{si } i = \ell \end{cases}$$

Théorème 1.11.2. (a) les colonnes $A_{B(i)}$, $i \neq \ell$ et A_j sont linéairement indépendantes et ainsi $\overline{B}$ est une matrice de base.

(b) $y = x + \theta^* d$ est une solution réalisable de base associée à la matrice de base $\overline{B}$.

Remarque 1.11.3. Puisque θ^* est strictement positif, la nouvelle solution réalisable de base $x + \theta^* d$ est différente de x ; puisque d est une direction de diminution du coût, alors le coût de cette nouvelle solution réalisable de base est plus petite que la précédente c'est-à-dire

$$c^T(x + \theta^* d) = c^T x + \theta^* \bar{c}_j < c^T x.$$

Cela signifie qu'on s'est déplacé vers une nouvelle solution réalisable de base avec un coût de la fonction objectif qui est plus petite que le précédente.

Nous pouvons ainsi résumer un type d'itération pour la méthode du simplexe.

Pour ce faire, nous allons définir le vecteur $u = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}$ en posant $u = -d_B = B^{-1}A_j$, A_j étant la colonne entrante dans la base. En particulier $u_i = -d_{B(i)}$ pour $i = 1, \dots, m$.

1.11.3 Une itération de l'algorithme du simplexe

1. On commence par une matrice de base $B = \begin{bmatrix} | & & | \\ A_{B(1)} & \dots & A_{B(m)} \\ | & & | \end{bmatrix}$ et une solution réalisable de base associée.
2. On calcule les coûts réduits $\bar{c}_j = c_j - c_B^T B^{-1} A_j$ pour toutes les variables hors base.
 - (a) Si tous sont positifs (≥ 0), la solution réalisable en cours est optimale et l'algorithme se termine.
 - (b) Sinon on choisit un indice j pour lequel $\bar{c}_j < 0$.
3. Calculer $u = B^{-1} A_j$. Si aucune composante de u n'est strictement positive, alors $\theta^* = +\infty$ et le coût optimal est $-\infty$ et l'algorithme prend fin.
4. Si certaines composantes de u sont strictement positives, alors on détermine

$$\theta^* = \min_{\{i=1, \dots, m | u_i > 0\}} \left(\frac{x_{B(i)}}{u_i} \right). \quad (1.11)$$

5. Soit ℓ l'indice tel que $\theta^* = \frac{x_{B(\ell)}}{u_\ell}$. On forme une nouvelle matrice de base $\overline{B}$ en remplaçant $A_{B(\ell)}$ par A_j . Si y est la nouvelle solution réalisable de base, alors les nouvelles valeurs des variables de base sont

$$y_j = \theta^* \text{ et } y_{B(i)} = x_{B(i)} - \theta^* u_i, \text{ avec } i \neq \ell.$$

Théorème 1.11.3. *On suppose que l'ensemble des réalisables n'est pas vide et que chaque solution réalisable de base est non dégénérée. Alors, la méthode du simplexe prend fin après un nombre fini d'itérations. À la fin de l'algorithme, les deux possibilités suivantes peuvent se présenter :*

- (a) *On obtient une base optimale B et une solution réalisable de base associée qui est optimale.*
- (b) *On trouve un vecteur d satisfaisant $Ad = 0$, $d \geq 0$ et $c^T d < 0$, et le coût optimal est $-\infty$.*

1.11.4 Remarques sur la méthode du simplexe pour les problème dégénérés

Jusqu'ici, nous avons développé l'algorithme du simplexe en supposons que toutes les solutions réalisables de base sont non dégénérées. Nous supposons maintenant le même algorithme est développé mais en présence au moins d'une solution réalisable dégénérée. Si la solution réalisable de base x en cours est dégénérée, alors $\theta^* = \min_{\substack{i=1,\dots,m \\ u_i > 0}} \frac{x_B(i)}{u_i} = \frac{x_B(\ell)}{u_\ell}$ peut être nul et donc, la nouvelle solution réalisable de base $y = X + \theta^* d$ sera la même que x bien que la nouvelle matrice de base $\bar{B}$ soit différente de B (B étant la matrice de base à laquelle x est associée). Cela se passe lorsque $x_B(\ell) = 0$ et $u_\ell = -d_{B(\ell)} > 0$. Dès lors, il se peut qu'on ait "**un phénomène indésirable**" qu'on appelle **phénomène cyclique** dans l'algorithme du simplexe. Ce phénomène consiste à retomber, dans le processus de l'algorithme, sur la matrice de base et la solution réalisable de base initiales. **En conclusion, dans le cas dégénéré, il n'est pas garanti que la méthode du simplexe se termine après un nombre fini d'itérations. Nous ferons un exemple dans la suite.**

1.11.5 Sélection du pivot

Dans la méthode du simplexe que nous avons décrite, à l'étape 2 nous sommes libre de choisir n'importe quel indice j dont le coût réduit $\bar{c}_j$ est strictement négatif ; de même, à l'étape 4, il peut y avoir plusieurs indices l qui atteignent le minimum dans la définition de θ^* , et nous sommes libre de choisir n'importe quel de ces indices. Les règles qui permettent de faire de tels choix sont appelées **règles du pivot**.

Concernant le choix de la colonne entrante, les règles suivantes sont possibles :

- (a) Choisir une colonne A_j , avec $\bar{c}_j < 0$, dont le coût réduit est le plus strictement inférieur à zéro (c'est-à-dire dont la valeur absolue du coût réduit est la plus grande).
- (b) Choisir une colonne avec $\bar{c}_j < 0$ pour lequel $\theta^* |\bar{c}_j|$ est le plus grand. Cette règle offre la possibilité d'arriver à l'optimalité après un tout petit nombre d'itérations. Mais son inconvénient est qu'à chaque itération, il faut calculer θ^* pour chacun des colonnes avec $\bar{c}_j < 0$: ce qui rend le travail un peu long.

Remarque 1.11.4. *Dans la pratique, certaines règles sont parfois utilisées. Par exemple, la règle du plus petit indice encore appelée règle de Bland qui consiste à choisir le plus petit indice pour lequel $\bar{c}_j < 0$. Sous cette règle, la colonne entrante est celle de la première variable hors base dont le coût réduit est strictement négatif.*

Concernant le choix de la colonne sortante, l'option la plus simple est encore celle de **la règle de Bland** : parmi toutes les variables éligibles de sortir de la base, choisir celle avec le plus petit indice. Il a été démontré qu'en utilisant la règle du plus petit indice, à la fois pour la colonne entrante et celle sortante, on peut **éviter le phénomène cyclique pour la méthode du simplexe dans le cas dégénéré**.

1.11.6 Algorithme du simplexe révisé

Selon l'algorithme du simplexe, au début de chaque itération on a besoin de calculer l'inverse B^{-1} de la nouvelle matrice de base qui intervient dans le calcul des coûts réduits et du vecteur $B^{-1}A_j$ qui jouent un rôle important dans le processus de l'algorithme. Pour ce faire, nous allons développer une méthode efficace pour mettre à jour l'inverse B^{-1} à chaque fois que nous effectuons un changement de base.

Soit

$$B = [A_{B(1)} \quad \dots \quad A_{B(m)}]$$

la matrice de base au début d'une itération et

$$\overline{B} = [A_{B(1)} \quad \dots \quad A_{B(\ell-1)} \quad A_{B(j)} \quad A_{B(\ell+1)} \quad \dots \quad A_{B(m)}] \quad (1.12)$$

la matrice de base de l'itération suivante. Ces deux matrices de base ont les mêmes colonnes sauf la ℓ -ième colonne $A_{B(\ell)}$ (colonne sortante de la base) qui est remplacée par A_j . Puisque $B^{-1}B = I$, on voit que $B^{-1}A_{B(i)}$ est le i -ème vecteur unitaire e_i . Ainsi, on a

$$B^{-1}\overline{B} = [e_{(1)} \quad \dots \quad e_{(\ell-1)} \quad u \quad e_{(\ell+1)} \quad \dots \quad e_{(m)}]$$

$$= \begin{pmatrix} 1 & & & u & & \\ & \ddots & & \vdots & & \\ & & & u_\ell & & \\ & & & \vdots & \ddots & \\ & & & u_m & & 1 \end{pmatrix}$$

où $u = B^{-1}A_j$. Utilisons une suite d'opérations élémentaires sur les lignes qui transformera la matrice précédente en la matrice identité. Pour obtenir la matrice identité, on peut alors effectuer les opérations élémentaires suivantes sur les lignes.

- (a) Pour tout $i \neq \ell$, nous ajoutons à la ligne i , la ligne ℓ multipliée par le quotient $-\frac{u_i}{u_\ell}$. ((rappelons que $u_\ell > 0$).). Cette opérations transforme u_i en zéro.
- (b) Nous divisons la ligne ℓ par ℓ . Cette opération transforme u_ℓ en un ("1").

Ainsi ces opérations transforme la colonne u au i -ème vecteur unitaire ℓ qui est e_ℓ . Cette suite d'opérations élémentaires sur les lignes est équivalente à multiplier à gauche la matrice $B^{-1}\overline{B}$ par une certaine matrice inversible Q . Ainsi on a $QB^{-1}\overline{B} = I$. Par conséquent $QB^{-1} = \overline{B}^{-1}$. Cette dernière formule de $\overline{B}^{-1}$ montre que si nous appliquons les mêmes suites d'opérations sur les lignes à la matrice B^{-1} (ce qui équivaut à multiplier à gauche B^{-1} par Q) nous obtenons $\overline{B}^{-1}$. Nous concluons que tout ce qu'il faut prendre pour générer $\overline{B}^{-1}$, est de commencer avec B^{-1} et d'appliquer la suite des opérations élémentaires sur les lignes décrite ci-dessus.

Exemple 16. Soit

$$B^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ -2 & 3 & 1 \\ 4 & -3 & -2 \end{pmatrix}, \quad u = \begin{pmatrix} -4 \\ 2 \\ 2 \end{pmatrix},$$

et supposons que $\ell = 3$. Notre objectif est de transformer le vecteur u au troisième vecteur unité $e_3 = (0, 0, 1)$. Nous multiplions la 3 ième ligne par 2 et l'ajoutons à la 1ère ligne. Nous soustrayons

la 3^{ème} ligne de la seconde ligne. Enfin, nous divisons la 3^{ème} ligne par 2. Nous obtenons

$$\overline{B}^{-1} = \begin{pmatrix} 9 & -4 & -1 \\ -6 & 6 & 3 \\ 2 & -1,5 & -1 \end{pmatrix}.$$

Si nous considérons la détermination de B^{-1} de la manière que nous avons décrite, nous obtenons une implémentation de la méthode du simplexe connue sous le nom de la méthode du simplexe révisée. En voici un résumé de la méthode du simplexe révisée.

1.11.7 Une itération de la méthode du simplexe révisée

1. On commence par une matrice de base $B = \begin{bmatrix} | & & | \\ A_{B(1)} & \dots & A_{B(m)} \\ | & & | \end{bmatrix}$, une solution réalisable de base associée x et l'inverse B^{-1} de la matrice de base.
2. On calcule le vecteur ligne $p^T = c_B^T B^{-1}$ et on calcule les coûts réduits $\bar{c}_j = c_j - p^T A_j$ pour toutes les variables hors base.
 - (a) Si tous sont positifs (≥ 0), la solution réalisable en cours est optimale et l'algorithme se termine.
 - (b) Sinon on choisit un indice j pour lequel $\bar{c}_j < 0$.
3. Calculer $u = B^{-1} A_j$. Si aucune composante de u n'est strictement positive, alors le coût optimal est $-\infty$ et l'algorithme prend fin.
4. Si certaines composantes de u sont strictement positives, alors on détermine

$$\theta^* = \min_{\{i=1,\dots,m \mid u_i > 0\}} \left(\frac{x_{B(i)}}{u_i} \right). \quad (1.13)$$

5. Soit ℓ l'indice tel que $\theta^* = \frac{x_{B(\ell)}}{u_\ell}$. On forme une nouvelle matrice de base $\overline{B}$ en remplaçant $A_{B(\ell)}$ par A_j . Si y est la nouvelle solution réalisable de base, alors les nouvelles valeurs des variables de base sont

$$y_j = \theta^* \text{ et } y_{B(i)} = x_{B(i)} - \theta^* u_i, \text{ avec } i \neq \ell.$$

6. Former la matrice $[B^{-1} \mid u]$ de type $m \times (m+1)$. Ajouter à chacune de ces ligne un multiple de la ligne ℓ pour transformer sa dernière colonne c'est-à-dire la colonne u en la ℓ i-ème vecteur unitaire e_ℓ . Ainsi les m premières colonnes issues de ces opérations est la matrice $\overline{B}^{-1}$.

1.11.8 Implémentation de l'algorithme du simplexe dans un tableau.

On considère le problème de programmation linéaire suivant sous forme standard :

$$\begin{array}{ll} \text{minimiser} & c^T x \\ \text{sous les contraintes} & Ax = b \\ & x \geq 0, \end{array}$$

Nous supposons que A est une matrice de dimensions $m \times n$ et que ses lignes sont linéairement indépendantes.

Pour résoudre ce problème, nous allons utiliser l'algorithme du simplexe. Cet algorithme est une itération qui nous conduit à déterminer si possible une solution optimale du problème.

La méthode du simplexe commence par la connaissance d'une matrice de base $B = [A_{B(1)} \dots A_{B(m)}]$ et de la solution de base réalisable x associée à cette matrice de base. On peut réaliser cette itération en implémentant la méthode du simplexe dans un tableau dont la structure est

$-c_B^T B^{-1}b$	$c^T - c_B^T B^{-1}A$
$B^{-1}b$	$B^{-1}A$

ou de façon détailler on a :

$-c_B^T B^{-1}b = -c_B^T x_B$	$\bar{c}_1$	$\dots$	$\bar{c}_n$
$x_{B(1)}$			
$\vdots$	$B^{-1}A_1$	$\dots$	$B^{-1}A_n$
$x_{B(m)}$			

Remarque 1.11.5. .

Dans le tableau,

- (i) l'expression $-c_B^T B^{-1}b$ représente l'opposé du coût ou de la valeur de la fonction objectif correspondant à la solution de réalisable en cour.
- (ii) la ligne contenant les coûts réduits $\bar{c}_i$ est appelée **la ligne zéro du tableau**.
- (iii) la ligne contenant $x_{B(i)}$ est la **la ième ligne du tableau**.
- (iv) la colonne contenant le vecteur des variables de base $x_B = B^{-1}b$ est appelée **la colonne zéro du tableau** ;
- (vi) la colonne $B^{-1}A_i$ est appelée **la ième colonne du tableau**.

1.11.9 Itération de l'algorithme du simplexe dans un tableau

1. Une telle itération commence avec un tableau qui est associé à la matrice de base B choisie et à la solution de base réalisable correspondante.
2. Examiner les coûts réduits de la ligne zéro du tableau. S'ils sont tous positifs, alors la solution de base réalisable en cour est optimale, et l'algorithme se termine ; dans le cas contraire, choisir un indice j pour lequel $\bar{c}_j < 0$.
3. Considérer le vecteur $u = B^{-1}A_j$, qui est la j ème colonne (colonne du pivot) du tableau. Si aucune composante de u n'est strictement positive, alors le coût optimal du problème est $-\infty$, et l'algorithme se termine.
4. Dans le cas contraire, pour chaque i pour lequel u_i est strictement positive, on calcul le ratio $\frac{x_{B(i)}}{u_i}$. Soit ℓ l'indice de la ligne qui correspond à la plus petite valeur de ces ratios

$\left(\text{c'est-à-dire } \theta^* = \min_{\substack{i=1,\dots,m \\ u_i > 0}} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell} \right)$ (cette ligne est appelée ligne du pivot). Dans ce cas, la colonne $A_{B(\ell)}$ sort de la base et la colonne A_j rentre dans la base. Ainsi on obtient une nouvelle base $\bar{B}$.

5. Ajouter à chaque ligne du tableau un multiple constant de la ℓ -ième ligne (ligne du pivot) pour que u_ℓ (l'élément pivot) devienne un (1) et les autres composantes de la colonne du pivot devienne zéro.

Remarque 1.11.6. .

- (i) *Lorsqu'on arrive à l'étape 5. de la méthode du simplexe, on obtient un nouveau tableau associé à la nouvelle base $\bar{B}$ dans lequel on peut lire la nouvelle solution de base réalisable. En effet, on obtient un tableau qui vérifie*

$-c_B^T \bar{B}^{-1} b = -c_B^T x_{\bar{B}}$	$\bar{c}_1$	$\dots$	$\bar{c}_n$
$x_{\bar{B}(1)}$	/		/
$\vdots$	$\bar{B}^{-1} A_1$	$\dots$	$\bar{B}^{-1} A_n$
$x_{\bar{B}(m)}$	/		/

Ici les $\bar{c}_j = c_j - c_B^T \bar{B}^{-1} A_j$, $j = 1, \dots, n$.

- (ii) **Détermination de $\bar{B}^{-1}$**

On forme la matrice $[B^{-1}|u]$ qui est une matrice de dimension $m \times (m+1)$. On ajoute à chaque ligne de cette matrice un multiple constant de la ℓ -ième ligne (ligne du pivot) pour que u_ℓ (l'élément pivot) devienne un (1) et que les autres composantes de la dernière colonne de cette matrice soient nulles. Ainsi les m premières colonnes de la matrice obtenue est la matrice $\bar{B}^{-1}$.

Théorème 1.11.4 (algorithme du simplexe dans le cas non dégénéré). *Supposons que la région des réalisables est non vide et que chaque solution de base réalisable soit non dégénérée. Alors, la méthode se termine après un nombre fini d'itérations. A la fin, les deux situations suivantes peuvent se présenter :*

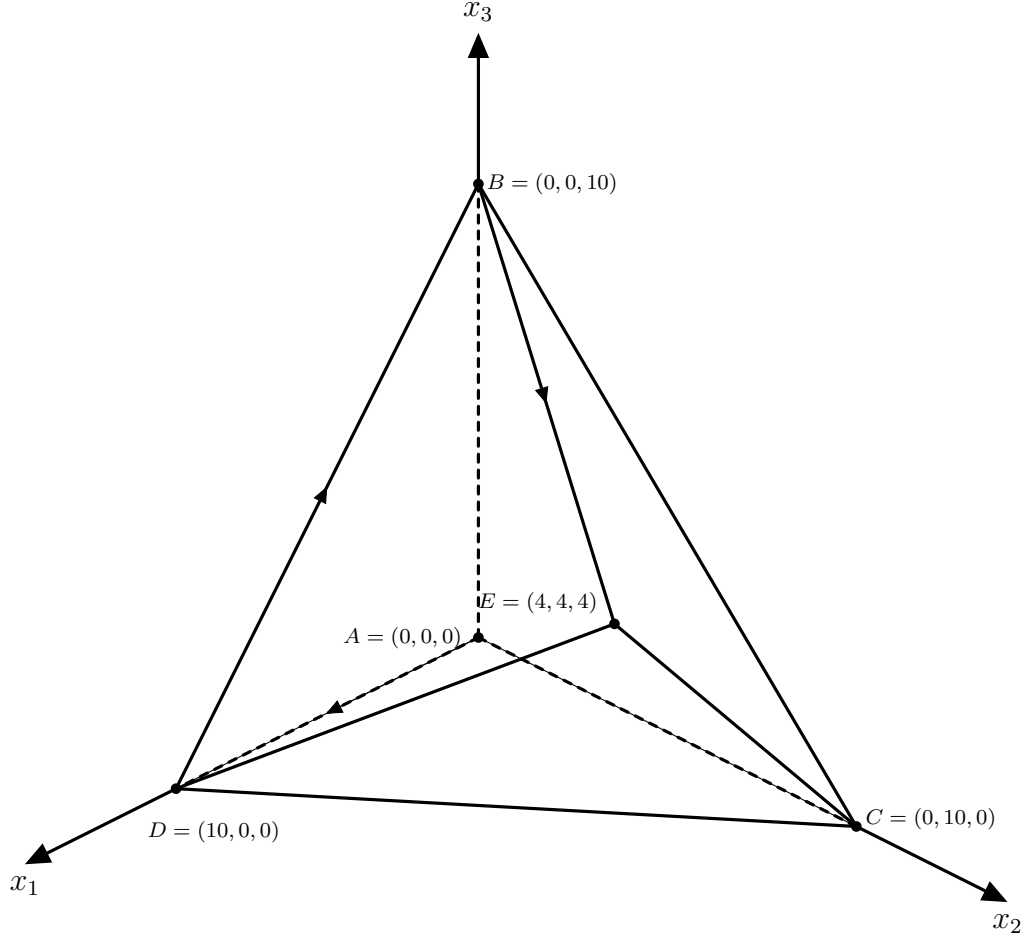
- (a) *Nous avons une base optimale B et une solution réalisable de base associée qui est optimale.*
- (b) *Nous avons une variable hors base x_j telle que son coût réduit $\bar{c}_j < 0$ et toutes les composantes u_i du vecteur $u = B^{-1} A_j$ sont telles que $u_i \leq 0$. Dans ce cas, le coût optimal est $-\infty$.*

Exemple 17. *À l'aide de l'algorithme du simplexe, déterminer la solution optimale du programme linéaire suivant. On précisera la valeur optimale de la fonction objectif et la matrice de base optimale.*

$$\begin{array}{llllll}
 \text{Min} & -10x_1 & - & 12x_2 & - & 12x_3 \\
 & x_1 & + & 2x_2 & + & 2x_3 & \leq & 20, \\
 (PL) & 2x_1 & + & x_2 & + & 2x_3 & \leq & 20, \\
 & 2x_1 & + & 2x_2 & + & x_3 & \leq & 20 \\
 & x_1, & x_2, & x_3 & \geq & 0 & .
 \end{array}$$

Résolution

L'ensemble des solutions réalisables de ce programme linéaire est représenté ci-dessous.



Pour appliquer l'algorithme du simplexe, il faut mettre ce programme linéaire sous forme standard. Nous introduisons donc les variables d'écart (x_4 , x_5 et x_6) pour obtenir la forme standard (FS) suivant :

$$\begin{array}{rcll}
 \text{Min} & -10x_1 & - & 12x_2 & - & 12x_3 & & & & \\
 (FS) & x_1 & + & 2x_2 & + & 2x_3 & + & x_4 & & = 20, \\
 & 2x_1 & + & x_2 & + & 2x_3 & & & + & x_5 & = 20, \\
 & 2x_1 & + & 2x_2 & + & x_3 & & & & + & x_6 & = 20 \\
 & x_1, & x_2, & x_3, & x_4, & x_5, & x_6 & \geq & 0 & & .
 \end{array}$$

La matrice des contraintes en terme d'inégalité de (FS) est

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

La matrice identité $B = [A_4 \ A_5 \ A_6] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ est une matrice de base extraite de A .

La solution réalisable de base associée à B est $x^T = (0, 0, 0, 20, 20, 20)$ car les variables hors base sont $x_1 = x_2 = x_3 = 0$ et $x_B = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix} = B^{-1}b = b = \begin{pmatrix} 20 \\ 20 \\ 20 \end{pmatrix}$.

Pour la matrice de base $B = [A_4 \ A_5 \ A_6]$ on a $c_B^T = (0, 0, 0)$. Ainsi $C_B^T x_B = 0$ et $\bar{c}^T = c^T = (-10, -12, -12, 0, 0, 0)$. Par conséquent le tableau initial de l'algorithme du simplexe est donné par :

		x_1	x_2	x_3	x_4	x_5	x_6
	0	-10	-12	-12	0	0	0
$x_4 =$	20	1	2	2	1	0	0
$x_5 =$	20	2*	1	2	0	1	0
$x_6 =$	20	2	2	1	0	0	1

Remarque 1.11.7. Les variables de base sont mentionnées sur le tableau à gauche. $x_i =$ à gauche du tableau nous indique quelles sont les variables de base et dans quel ordre elles sont. Par exemple la première variable de base est x_4 et sa valeur est 20. La deuxième variable de base est $x_5 = 20$ et la troisième est $x_6 = 20$. Normalement, il n'est pas nécessaire de mentionner $x_4 =$, $x_5 =$, et $x_6 =$. En effet la colonne dans le tableau associée à la première variable de base doit être le premier

vecteur unitaire c'est-à-dire $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. Comme nous pouvons le voir dans le tableau, la colonne qui est associée à la variable x_4 est le premier vecteur unitaire, ce qui implique que la première variable de base est bien x_4 . C'est ainsi qu'on peut déterminer la deuxième variable de base, la troisième variable de base et ainsi de suite.

Nous continuons avec notre exemple.

Le coût réduit de la variable x_1 est négatif. Nous pouvons donc faire entrer la variable x_1 dans la base. Dans ce cas la colonne pivot est la colonne de x_1 c'est-à-dire $u = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$. On détermine le

minimum $\min \left\{ \frac{x_4}{u_1}, \frac{x_5}{u_2}, \frac{x_6}{u_3} \right\} = 10$. Ce minimum est atteint pour la deuxième et la troisième ligne du tableau. Nous choisissons alors la deuxième ligne du tableau comme la ligne pivot. Ainsi le pivot est l'élément que nous avons mentionné avec l'astérisque. Dans ce cas la variable x_5 sort de la base (elle est remplacée par x_1). Nous obtenons alors une nouvelle base qui $\bar{B} = [A_4 \ A_1 \ A_6]$. Pour continuer l'itération de l'algorithme du simplexe il faut aller au tableau suivant car la solution réalisable de base associée à la base B n'est pas optimale. Dans le nouveau tableau, n doit avoir des zéros partout sur la colonne du pivot sauf à la place du pivot où il faut avoir 1. Ainsi nous effectuons les différentes opérations suivantes :

- nous multiplions la ligne du pivot par 5 et nous l'ajoutons à la ligne zéro ;
- nous multiplions la ligne du pivot par $-\frac{1}{2}$ nous l'ajoutons à la première ligne ;
- nous multiplions la ligne du pivot par -1 nous l'ajoutons à la troisième ligne du tableau.
- nous divisons la ligne du pivot par le pivot.

Ces opérations nous conduisent alors à un nouveau tableau :

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	10	0	1,5	1*	1	-0,5	0
$x_1 =$	10	1	0,5	1	0	0,5	0
$x_6 =$	0	0	1	-1	0	-1	1

Remarque 1.11.8. La solution réalisable de base correspondant à ce tableau est $x^T = (10, 0, 0, 10, 0, 0)$. Cette solution est associée à la base $\bar{B} = [A_4 \ A_1 \ A_6]$. Notons que cette solution est dégénérée car la variable de base $x_6 = 0$. La valeur de la fonction objectif correspondante à cette solution est -100. Nous déduisons de x^T que $(10, 0, 0)$ est une solution réalisable du problème original. La solution $x^T = (10, 0, 0, 10, 0, 0)$ associée à la base $\bar{B} = [A_4 \ A_1 \ A_6]$ n'est pas optimale.

Nous continuons notre exemple. Dans le présent tableau, les variables x_2 et x_3 ont des coûts réduits négatifs. Choisissons la variable x_3 comme la variable entrante dans la base. La colonne du pivot est alors la colonne de x_3 c'est-à-dire $(1, 1, -1)$.

$\min \left\{ \frac{x_4}{u_1}, \frac{x_1}{u_2} \right\} = \min \{10, 10\} = 10$. Le minimum est atteint pour la première ligne c'est-à-dire la ligne de x_4 et pour la deuxième ligne c'est-à-dire la ligne de x_1 . Nous allons choisir la première ligne du tableau comme étant la ligne du pivot. Donc le pivot est alors 1 que nous avons mis en relief par l'astérisque. Nous obtenons alors le tableau suivant après avoir effectué les opérations nécessaires sur les lignes du présent tableau.

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_3 =$	10	0	1,5	1	1	-0,5	0
$x_1 =$	0	1	-1	0	-1	1	0
$x_6 =$	10	0	2,5*	0	1	-1,5	1

La solution correspondant à ce tableau est $x^T = (0, 0, 10, 0, 0, 10)$. Cette solution est une solution de base réalisable et dégénérée. Elle est associée à la base $B = [A_3, A_1, A_6]$. La valeur de la fonction objectif correspondante est -120. Le coût réduit de la variable x_2 est négatif. Donc la variable x_2 entre dans la base et la variable x_6 sort de la base. Nous obtenons ainsi le tableau suivant :

		x_1	x_2	x_3	x_4	x_5	x_6
	136	0	0	0	3,6	1,6	1,6
$x_3 =$	4	0	0	1	0,4	0,4	-0,6
$x_1 =$	4	1	0	0	-0,6	0,4	0,4
$x_2 =$	4	0	1	0	0,4	-0,6	0,4

Tous les coûts réduits sont positifs dans ce tableau. Donc la solution correspond à ce tableau est une solution optimale du problème sous forme standard. Cette solution est $x_{opt}^T = (4, 4, 4, 0, 0, 0)$. La valeur optimale de la fonction objectif est alors $v_{opt} = -136$. La matrice de base optimale est alors $B = [A_3 \ A_1 \ A_2]$.

En définitive, la solution optimale du problème original est alors $x_{opt}^{*T} = (4, 4, 4)$. et la valeur optimale de la fonction objectif est la même que pour la forme standard c'est-à-dire -136.

1.11.10 Exemple de l'algorithme du simplexe pour les problème dégénérés

Dans l'exemple qui suit nous présentons un tableau de l'algorithme du simplexe où une solution réalisable de base est dégénérée. Dans cet exemple, dans un premier temps, nous allons montrer l'existence du phénomène cyclique en utilisant les règles suivantes pour le choix du pivot :

- (a) la colonne entrante dans la base est celle dont le coût réduit est le plus strictement négatif;
- (b) parmi toutes les colonnes éligibles de sortir de base, on choisit la première.

Dans un second temps, nous allons la solution optimale en suivant la règle de Bland.

Exemple 18. *Considérons un problème décrit par le tableau initial suivant :*

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	$-3/4$	20	$-1/2$	6	0	0	0
$x_5 =$	0	$1/4^*$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

1.11.11 Méthode du simplexe en deux phases

Pour démarrer la méthode du simplexe, nous devons trouver une première solution de base réalisable. Parfois, c'est simple à trouver. Par exemple, supposons que nous voulons résoudre un programme linéaire ayant pour contraintes

$$\begin{cases} Ax \leq b \\ x \geq 0, \end{cases}$$

où $b \geq 0$. Nous pouvons alors introduire des variables d'écart positives $s = (s_1, \dots, s_m)$ (m est le nombre de lignes de A) et réécrire les contraintes sous la forme

$$\begin{cases} Ax + s = b \\ x, s \geq 0, \end{cases}$$

Dans ce cas le vecteur (x, s) défini par $x = 0$ et $s = b$ est une solution de base réalisable et la matrice de base correspondante est l'identité. Cependant, en général, trouver une solution réalisable de base initiale n'est pas facile et cela nécessite, comme nous le verrons brièvement, la solution d'un problème de programmation linéaire auxiliaire. Dans cette section nous allons utiliser la méthode du simplexe à deux phase pour trouver si possible une solution réalisable de base d'un problème sous forme standard et ensuite de résoudre éventuellement le problème.

Considérons le problème

$$(FS) \begin{cases} \min & c^T x \\ \text{sous les contraintes} & Ax = b \\ & x \geq 0, \end{cases}$$

à résoudre par la méthode du simplexe à deux phases.

PHASE I :

Cette phase a pour objectif de trouver une solution réalisable de base des contraintes de (FS) ou de décider de l'impossibilité de la résolution de (FS). Les étapes de la phase I sont les suivantes :

1. Transformer éventuellement le problème (FS) de sorte que $b \geq 0$ en multipliant certaines de ses contraintes par -1 .

2. Résoudre le problème auxiliaire suivant, associé à (FS) , par la méthode du simplexe.

$$(PA) \begin{cases} \min & Z = y_1 + y_2 + \dots + y_m \\ \text{sous les contraintes} & Ax + y = b \\ & x, y \geq 0, \end{cases}$$

où $y = (y_1, y_2, \dots, y_m)$ est un vecteur dont les composantes y_i sont appelées variables artificielles.

La résolution du problème auxiliaire (PA) commence par la solution réalisable de base $(x, y) = (0, b)$ dont la matrice de base est la matrice identité.

3. À l'issue de la résolution de (PA) , si son coût optimal $Z_{opt} > 0$, alors le programme linéaire (FS) n'est pas résoluble. Dans ce cas on arrête la résolution de (FS) .
4. Si $Z_{opt} = 0$ et il n'y a aucune variable artificielle dans les composantes du vecteur de base correspondant à la solution optimale de (PA) , alors une solution réalisable de base initiale du problème (FS) est trouvée.
5. Sinon, tous les $y_i^* = 0$ (car $Z_{opt} = 0$), mais certaines sont encore des variables de base de la solution optimale du problème auxiliaire (PA) . Sans perte de généralités, supposons $A_{B(1)}, A_{B(2)}, \dots, A_{B(k)}$ sont les colonnes de A qui se trouvent dans la matrice de base du dernier tableau de simplexe de (PA) . Soit $y_{k+1}^* = y_{k+2}^* = \dots = y_m^* = 0$ les variables artificielles dans les variables de base de la solution optimale du problème auxiliaire (PA) .
6.
 - Supposons qu'il existe $y_i^* = 0$, $i \in \{k+1, k+2, \dots, m\}$ une variable artificielle dans les variables de base tel qu'il existe une colonne $B^{-1}A_j$ du dernier tableau de simplexe de (PA) dont la i -ème composante $u_{ij} \neq 0$. Alors il faut sortir la colonne correspondant à y_i^* de la base et faire rentrer la colonne A_j de la matrice A dans la base. Ceci est accompli de la manière habituelle : effectuez les opérations élémentaires sur les lignes de façon à transformer la colonne $B^{-1}A_j$ par le i -ème vecteur unitaire. La seule différence avec la mécanique habituelle de la méthode du simplexe est que l'élément pivot ($u_{ij} \neq 0$, la i -ème composante de $B^{-1}A_j$) pourrait être négatif. Étant donné que la i -ème variable de base était nulle, l'ajout d'un multiple de la i -ème ligne aux autres lignes du tableau ne modifie pas les valeurs des variables de base. Cela signifie qu'après le changement de base, nous en sommes toujours à la même solution réalisable de base du problème auxiliaire, mais nous avons réduit d'une unité le nombre de variables artificielles dans les variables de base. Nous répétons cette procédure autant de fois que nécessaire jusqu'à ce que toutes les variables artificielles soient éliminées de la base.
 - Cas où il existe une variable artificielle $y_i^* = 0$ dans les variables de base, $i \in \{k+1, k+2, \dots, m\}$, telle que la i -ème ligne de $B^{-1}A$ soit nulle. Alors cette ligne est redondante c'est-à-dire qu'elle correspond à une contrainte redondante (qui est combinaison linéaire d'autres contraintes). Dans ce cas on la supprime du tableau. Et Ainsi on poursuit le processus tant que les conditions sont vérifiées jusqu'à ce qu'on obtienne une matrice de base constituée uniquement des colonnes de A et la solution réalisable de base correspondante. À ce point on supprime aussi les colonnes des variables artificielles dans le tableau.

Phase II

1. La matrice de base finale et le tableau de simplexe obtenu de la phase I sont respectivement la matrice de base initiale et le premier tableau de simplexe de la phase II.

2. Calculer les coûts réduits de toutes les variables pour cette base initiale, en utilisant les coefficients de la fonction objectif du problème initial (FS).
3. Appliquer la méthode du simplexe pour résoudre alors le problème original.

Exemple 19. *En utilisant l'algorithme du simplexe à deux phases, résoudre le programme suivant*

$$\begin{array}{llllllll}
 \text{Min} & x_1 & + & x_2 & + & x_3 & & \\
 \text{S.C} & x_1 & + & 2x_2 & + & 3x_3 & & = 3, \\
 & -x_1 & + & 2x_2 & + & 6x_3 & & = 2, \\
 & & & 4x_2 & + & 9x_3 & & = 5 \\
 & & & & & + & 3x_3 & + x_4 = 1 \\
 & x_1, & x_2, & x_3, & x_4 & \geq & 0 &
 \end{array}$$

1.12 Dualité en programmation linéaire

Dans ce chapitre, nous commençons par un problème de programmation linéaire, appelé primal, et introduisons un autre problème de programmation linéaire, appelé le dual. La théorie de la dualité traite de la relation entre ces deux problèmes et révèle la structure plus profonde de la programmation linéaire. Il s'agit d'un outil théorique puissant qui a de nombreuses applications, fournit de nouvelles connaissances géométriques, et conduit à un autre algorithme de programmation linéaire (la méthode duale du simplexe).

1.12.1 Motivation

La théorie de la dualité peut être motivée par la méthode du multiplicateur de Lagrange, souvent utilisée en calcul pour minimiser une fonction soumise à des contraintes exprimées en terme d'égalité. Par exemple, pour résoudre le problème

$$\begin{cases} \min & x^2 + y^2 \\ \text{s.c} & x + y = 1, \end{cases}$$

nous introduisons le multiplicateur de Lagrange p et formons le lagrangien $L(x, y, p)$ défini par

$$L(x, y, p) = x^2 + y^2 + p(1 - x - y).$$

Tout en gardant p fixe, nous minimisons le Lagrangien sur tous les x et y , sous aucune contrainte, ce qui peut être fait en fixant $\frac{\partial L}{\partial x}$ et $\frac{\partial L}{\partial y}$ à zéro. La solution optimale à ce problème sans contrainte est

$$x = y = \frac{p}{2},$$

et dépend de p . La contrainte $x + y = 1$ nous donne la relation supplémentaire $p = 1$, et la solution optimale au problème original est $x = y = \frac{1}{2}$.

L'idée principale de l'exemple ci-dessus est la suivante. Au lieu de faire respecter la contrainte $x + y = 1$, nous permettons qu'elle soit violée et associons un multiplicateur de Lagrange, ou prix, p avec la quantité $1 - x - y$ par lequel elle est violée. Cela conduit à la minimisation sans contrainte de $x^2 + y^2 + p(1 - x - y)$. Lorsque le prix est correctement choisi ($p = 1$, dans notre exemple), la solution optimale au problème avec contrainte est également optimale pour le problème sans contrainte. En particulier, sous cette valeur spécifique de p , la présence ou l'absence de la contrainte

n'affecte pas le coût optimal.

La situation en programmation linéaire est similaire : nous associons une variable prix à chaque contrainte et commençons à rechercher les prix sous lesquels la présence ou l'absence des contraintes n'affecte pas le coût optimal. Il s'avère que les prix qui conviennent à cela peuvent être trouvés en résolvant un nouveau problème de programmation linéaire, appelé le dual de l'original. Nous avons ainsi motivé la formation du problème dual. Considérons le problème standard

$$\begin{array}{ll} \text{minimiser} & c^T x \\ \text{sous les contraintes} & Ax = b \\ & x \geq 0, \end{array}$$

que nous appelons le problème primal, et supposons qu'il possède une solution optimale x^* . Nous introduisons un problème beaucoup plus relaxé dans lequel la contrainte $Ax = b$ est remplacée par $p^T(b - Ax)$, où p est un vecteur prix de même dimension que b . Nous faisons face alors au problème

$$\begin{array}{ll} \text{minimiser} & c^T x + p^T(b - Ax) \\ \text{sous les contraintes} & x \geq 0. \end{array}$$

Soit $g(p)$ le coût optimal du problème relaxé qui est une fonction du vecteur prix p . Le problème relaxé permet plus d'options que celles présentes dans le problème primal, et nous nous attendons à ce que $g(p)$ ne soit pas supérieur au coût primal optimal $c^T x^*$. En effet,

$$g(p) = \min_{x \geq 0} [c^T x + p^T(b - Ax)] \leq c^T x^* + p^T(b - Ax^*) = c^T x^*,$$

car x^* est solution optimale du problème primal et donc satisfait à $Ax^* = b$. Ainsi, chaque p conduit à une borne inférieure $g(p)$ pour le coût optimal $c^T x^*$. Autrement, $\forall p, g(p) \leq c^T x^*$.

Le problème

$$\begin{array}{ll} \text{maximiser} & g(p) \\ \text{sous aucune contrainte} & \end{array}$$

peut alors être interprété comme une recherche de la borne inférieure $g(p)$ la plus proche possible de $c^T x^*$, et est connu sous le nom de problème dual. Le principal résultat de la théorie de la dualité affirme que le coût optimal dans le problème dual est égal au coût optimal $c^T x^*$ dans le primal. En d'autres termes, si les prix sont choisis tels que le vecteur prix p est une solution optimale pour le problème dual, l'option de violer les contraintes $Ax = b$ n'a aucune valeur.

En utilisant la définition de $g(p)$, on a

$$\begin{aligned} g(p) &= \min_{x \geq 0} [c^T x + p^T(b - Ax)] \\ &= p^T b + \min_{x \geq 0} (c^T - p^T A)x. \end{aligned}$$

Il faut noter que

$$\min_{x \geq 0} (c^T - p^T A)x = \begin{cases} 0, & \text{si } c^T - p^T A \geq 0^T \\ -\infty, & \text{sinon.} \end{cases}$$

Pour la maximisation de $g(p)$, on a juste besoin de considérer les valeurs de p pour lesquelles $g(p)$ n'est pas égal à $-\infty$. Ainsi on conclut que le problème dual est le même que le programme linéaire

$$\begin{array}{ll} \text{maximiser} & p^T b \\ \text{sous les contraintes} & p^T A \leq c^T. \end{array}$$

1.12.2 problème dual

1.12.3 Forme générale d'un problème dual

Soit A une matrice avec les lignes a'_i et les colonnes A_j . On donne, comme cela est indiqué ci-dessous à gauche, le problème de minimisation appelé le problème Primal. Le problème de maximisation à droite du problème de minimisation est appelé le problème Dual du problème primal

minimiser	$c^T x$	maximiser	$p^T b$
sous les contraintes	$a'_i x \geq b_i, \quad i \in M_1,$ $a'_i x \leq b_i, \quad i \in M_2,$ $a'_i x = b_i, \quad i \in M_3,$ $x_j \geq 0, \quad j \in N_1,$ $x_j \leq 0, \quad j \in N_2,$ $x_j \text{ quelconque } j \in N_3,$	sous les contraintes	$p_i x \geq 0, \quad i \in M_1,$ $p_i \leq 0, \quad i \in M_2,$ $p_i \text{ qcq}, \quad i \in M_3,$ $pa'_j \leq c_j, \quad j \in N_1,$ $pa'_j \geq c_j, \quad j \in N_2,$ $pa'_j = c_j \quad j \in N_3.$

Exemple 20.

<i>minimiser</i>	$x_1 + 2x_2 + 3x_3$	<i>maximiser</i>	$5p_1 + 6p_2 + 4p_3$
<i>sous les contraintes</i>	$-x_1 + 3x_2 = 5,$ $2x_1 - x_2 + 3x_3 \geq 6,$ $x_3 \leq 4,$ $x_1 \geq 0,$ $x_2 \leq 0,$ $x_3 \text{ quelconque},$	<i>sous les contraintes</i>	$p_1 \text{ qcq},$ $p_2 \geq 0,$ $p_3 \leq 0,$ $-p_1 + 2p_2 \leq 1,$ $3p_1 - p_2 \geq 2,$ $3p_2 + p_3 = 3.$

Nous pouvons résumer le passage du primal au dual dans le tableau suivant

PRIMAL	min	max	DUAL
Contraintes	$\geq b_i$	≥ 0	variables
	$\leq b_i$	≤ 0	
	$= b_i$	quelconque	
variables	≥ 0	$\leq c_j$	contraintes
	≤ 0	$\geq c_j$	
	quelconque	$= c_j$	

Théorème 1.12.1. *Si on transforme le dual en un problème de minimisation équivalent puis on forme ensuite le dual de ce dernier, nous obtenons un problème équivalent au problème original.*

Ce théorème peut-être simplifié de la façon suivante.

Théorème 1.12.2. *Le dual du dual est le primal*

Exemple 21. *Considérons l'exemple 20 et vérifions que le dual du problème dual est le problème primal de départ.*

Dual d'un problème sous forme matricielle

$$\begin{array}{ll}
\begin{array}{ll}
\text{minimiser} & c^T x \\
\text{sous les contraintes} & Ax = b \\
& x \geq 0,
\end{array}
&
\begin{array}{ll}
\text{maximiser} & p^T b \\
\text{sous les contraintes} & pa^T \leq c^T
\end{array} \\
\\
\begin{array}{ll}
\text{minimiser} & c^T x \\
\text{sous les contraintes} & Ax \geq b
\end{array}
&
\begin{array}{ll}
\text{maximiser} & p^T b \\
\text{sous les contraintes} & pa^T = c^T \\
& p \geq 0
\end{array}
\end{array}$$

1.12.4 théorèmes fondamentaux de la dualité

Théorème 1.12.3 (théorème faible de la dualité). *Si x est une solution possible du primal et p une solution possible du dual, alors $p^T b \leq c^T x$.*

Le théorème faible de la dualité permet de donner quelques informations utiles concernant les relations entre le primal et le dual. Nous obtenons alors les corollaires suivants

Corollaire 1.12.1. (a) *Si le coût optimal du primal est $-\infty$, alors le problème dual n'est pas résolvable.*

(b) *Si le coût optimal du dual est $+\infty$, alors le problème primal n'est pas résolvable.*

Corollaire 1.12.2. *Soit x et p les solutions possibles respectives du primal et du dual. Si $p^T b = c^T x$, alors x et p sont respectivement les solutions optimales du primal et du dual.*

Le théorème suivant est résultat important dans la théorie de la dualité de l'optimisation linéaire.

Théorème 1.12.4 (théorème fort de la dualité). *Si un problème de programmation linéaire a une solution optimale, alors il en est de même pour le problème dual et le coût optimal du primal et celui du dual sont égaux.*

1.12.5 Relation entre le primal et le dual

Dual \ Primal	coût optimal fini	coût non borné	irrésolvable
coût optimal fini	*		
coût non borné			*
irrésolvable		*	*

Le cas où les deux problèmes peuvent ne pas avoir de solutions peut en effet se réaliser comme le montre l'exemple ci-dessous.

Exemple 22. *Considérons le problème primal qui n'a pas de solutions*

$$\begin{array}{ll}
\text{minimiser} & x_1 - x_2 \\
\text{sous les contraintes} & x_1 + x_2 \geq 1, \\
& -x_1 - x_2 \geq 1.
\end{array}$$

Son dual est

$$\begin{array}{ll} \text{maximiser} & y_1 + y_2 \\ \text{sous les contraintes} & y_1 - y_2 = 1, \\ & y_1 - y_2 = 0, \\ & y_1, y_2 \geq 0. \end{array}$$

et n'admet pas aussi de solutions.

1.12.6 Écarts complémentaires

Une importante relation entre les solutions optimales du primal et du dual est donnée par les conditions des écarts complémentaires dans le théorème suivant.

Théorème 1.12.5 (Écarts complémentaires). *Soient x et p des solutions possibles respectives des problèmes primal et dual. Les vecteurs x et p sont des solutions optimales respectivement pour les deux problèmes si, et seulement si,*

$$\begin{aligned} p_i(a_i^T x - b_i) &= 0, \quad \forall i \\ x_j(c_j - pa_j^T) &= 0, \quad \forall j \end{aligned}$$

Remarque 1.12.1. *La première condition d'écart complémentaire est automatiquement vérifiée par toute solution possible d'un problème sous forme standard. Si le problème primal n'est pas sous forme standard et a une contrainte sous la forme $a_i^T x \geq b_i$, la condition d'écart complémentaire correspondante dit que la variable dual p_i est nulle à moins que la contrainte est active.*

Si le problème primal est sous forme standard et une solution optimale non dégérée est connue, alors les conditions d'écart complémentaire détermine une unique solution optimale pour le problème dual. Nous illustrons cela par l'exemple suivant.

CHAPITRE 2

THÉORIE DES GRAPHS

2.1 Définition d'un graphe

Un graphe G est un couple (V, E) de deux ensembles finis V et E . Les éléments de V sont appelés sommets et ceux de E sont appelés arêtes ou arcs (selon qu'on soit avec un graphe non orienté ou un graphe orienté que nous verrons dans la suite) du graphe. Une arête ou un arc du graphe est une relation qui lie deux sommets de ce graphe. Il existe des graphes non orientés et des graphes orientés.

2.2 Graphe non orienté

2.2.1 Définitions

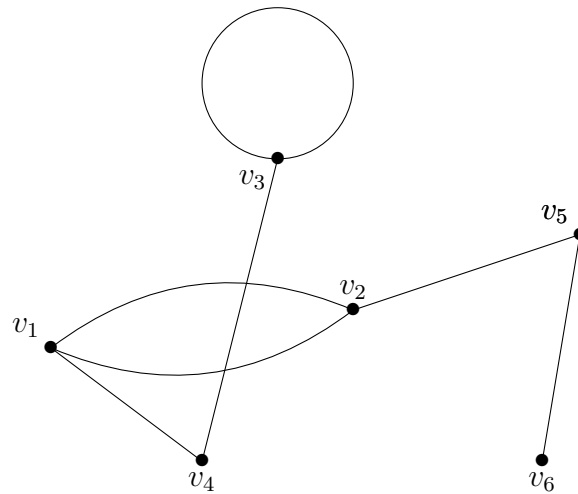
Soit $G = (V, E)$ un graphe où $V = \{v_1, \dots, v_n\}$ et $E = \{e_1, \dots, e_m\}$ sont deux ensembles finis. Les éléments de V sont les sommets. G est dit non orienté si chaque élément e de E est une paire (ensemble de deux points non ordonnés) de sommets a et b de V c-à-d $e = \{a, b\}$. Les éléments de E sont appelés arêtes. $e = \{a, b\}$ se lit aussi e relie les sommets a et b . Si l'arête e relie a et b , alors on dit que a et b sont adjacents ou incidents à e ou que l'arête e est incidente aux sommets a et b . Si $e = \{a, b\}$, on dit aussi que a et b sont les extrémités de e . Deux arêtes sont adjacentes si elles ont au moins une extrémité en commun.

- Un sommet auquel aucune arête n'est incidente est dit **isolé**.
- Lorsque les extrémités d'une arête sont le même sommet, alors cette arête est appelée **boucle**.
- **L'ordre** d'un graphe est le nombre $n = |V|$ ($|V|$ signifie cardinal de V) de sommets de ce graphe.

2.2.2 Représentation graphique d'un graphe non orienté

Les graphes tirent leur nom du fait qu'on peut les représenter par des dessins. Un graphe est souvent représenté dans le plan par des points qui sont les sommets reliés par des lignes qui sont les arêtes. Les arêtes ne sont pas forcément rectilignes.

Exemple 23. *exemple de graphe non orienté*

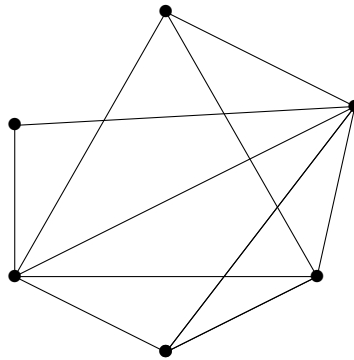


1. C'est un graphe non orienté $G = (V, E)$ ayant six sommets et sept arêtes. L'ensemble des sommets est $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ et l'ensemble des arêtes est $E = \{\{v_1, v_2\}, \{v_1, v_2\}, \{v_1, v_4\}, \{v_2, v_4\}, \{v_2, v_5\}, \{v_5, v_6\}, \{v_3, v_3\}\}$.
2. Dans ce graphe il y'a une boucle qui est l'arête $\{v_3, v_3\}$ et deux arêtes parallèles $\{v_1, v_2\}$ (Dans la représentation des graphes, on autorise qu'il y ait des répétitions d'arêtes entre deux sommets. Dans ce cas on parle d'arêtes parallèles.)

2.2.3 Quelques types de Graphes

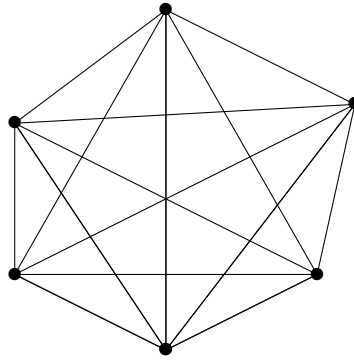
- Un graphe est **simple** s'il y'a au plus une arête qui relie deux sommets distincts et s'il n'y a pas de boucle sur un sommet.

Exemple 24. Le graphe suivant est un graphe simple



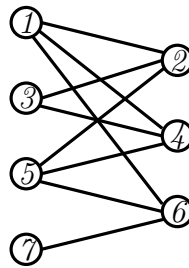
- Un graphe est **complet** si chaque sommet du graphe est relié à tous les autres sommets.

Exemple 25. Le graphe suivant est un graphe complet



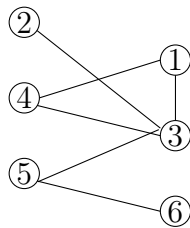
- Un graphe est dit **biparti** si l'ensemble des sommets peut être partitionné en deux parties X et Y de sorte que chaque arête du graphe relie un sommet dans X à un sommet dans Y .

Exemple 26. *Graphe biparti*

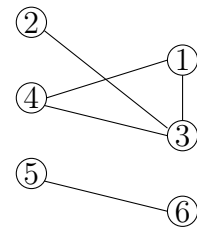


$V = \{1, 2, 3, 4, 5, 6, 7\}$ et $E = \{\{1, 2\}, \{1, 4\}, \{1, 6\}, \{3, 2\}, \{3, 4\}, \{5, 2\}, \{5, 4\}, \{5, 6\}, \{7, 6\}\}$
Ici $X = \{1, 3, 5, 7\}$ et $Y = \{2, 4, 6\}$

- Un graphe est **connexe** s'il est possible, à partir de n'importe quel sommet, de rejoindre tous les autres sommets en suivant les arêtes. Un graphe non connexe se décompose en composantes connexes.



Graphe connexe



Graphe non connexe

Composantes connexes : $V = \{1, 2, 3, 4\}$ et $V = \{5, 6\}$

2.3 Graphe orienté

2.3.1 Définitions

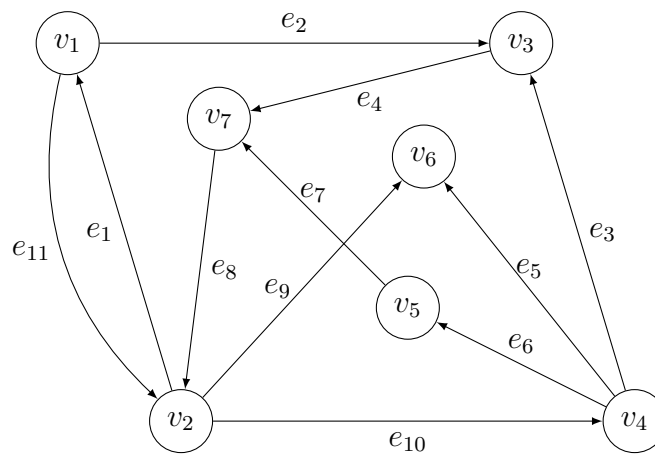
Un graphe $G = (V, E)$ est dit orienté si V est l'ensemble fini des sommets et E est un ensemble fini dont chaque élément e est un couple d'éléments de V . Un élément de E est appelé **arc**. Un arc e de l'ensemble E est défini par une paire ordonnée de sommets. Si $e = (a, b)$ est un arc du graphe G avec $a, b \in V$, alors on dit que a est l'extrémité initiale ou l'origine et b est l'extrémité finale ou la cible ou la destination de l'arc e . On dit aussi que le sommet a est un prédécesseur de b ou b

est un successeur de a . Si $e = (a, b) \in E$ est un arc du graphe, alors on dit que les sommets a et b sont adjacents ou que l'arc $e = (a, b)$ est incident aux sommets a et b . Lorsque l'extrémité initiale d'un arc est égale son extrémité finale, alors cet arc est une boucle. L'ensemble des arcs d'origine un sommet a est noté $\delta^+(a)$ et l'ensemble des arcs d'extrémité finale un sommet b est noté $\delta^-(b)$.

2.3.2 Représentation graphique d'un graphe orienté

Tout comme un graphe non orienté, on peut représenter un graphe orienté graphiquement. Dans ce cas, un sommet est représenté un point par et un arc par une flèche orientée de l'origine vers l'extrémité finale.

Exemple 27. *Le graphe orienté suivant possède 7 sommets et 11 arcs.*



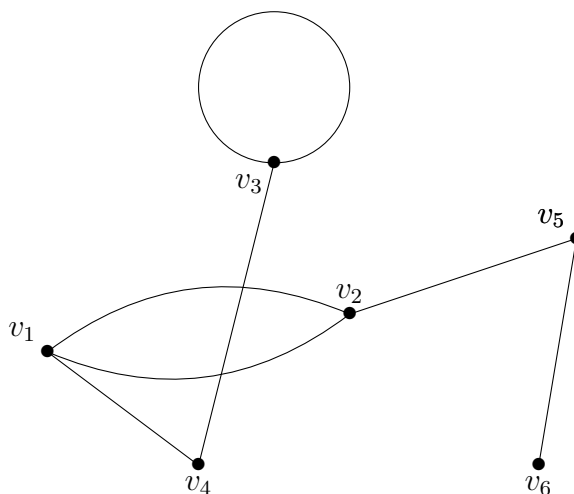
2.4 Degré d'un sommet-degré d'un graphe

- Le **degré d'un sommet** s , noté $\deg(s)$, est le nombre d'arêtes ou d'arcs incidents au sommet s .

Attention : Une boucle sur un sommet compte double.

- Le **degré d'un graphe** G est le degré maximum de tous ses sommets.
- Un graphe est **régulier** si tous ses sommets ont le même degré. Si le degré commun est k , alors on dit que le graphe est k -régulier.

Exemple 28. *:degré d'un graphe*



$$\deg(v_2) = 3, \deg(v_3) = 3, \deg(v_5) = 2 \text{ et } \deg(G) = 3.$$

Théorème 2.4.1. *La somme des degrés des sommets d'un graphe $G = (V, E)$ (orienté ou non) est égale $2 \times |E|$.*

Corollaire 2.4.1. *Dans un graphe (orienté ou non), le nombre de sommets de degré impair est pair.*

2.5 Chemin-circuit

- Un **chemin** dans un graphe orienté est une suite de la forme

$$(v_0, e_1, v_1, e_2, \dots, v_{k-1}, e_k, v_k)$$

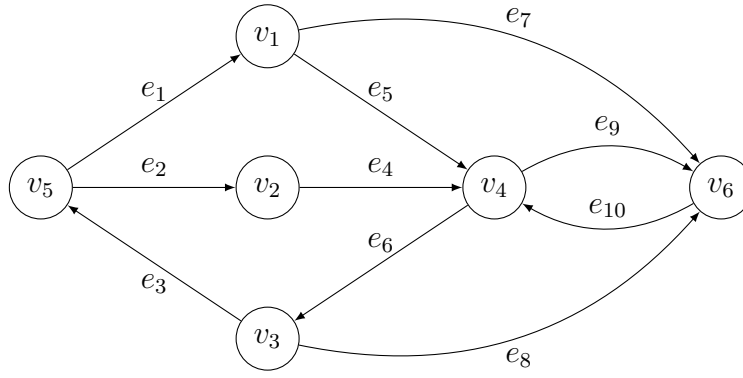
où k est un entier ≥ 0 , $v_i \in V$ pour $i = 0, \dots, k$ et $e_j \in E$ pour $j = 1, \dots, k$ avec $e_j = (v_{j-1}, v_j)$. En d'autres termes, un chemin est un trajet possible dans la représentation d'un graphe orienté lorsqu'on suit les arcs dans le sens des flèches. L'entier k qui représente le nombre d'arcs dans le chemin est appelé la **longueur du chemin**.

- Un chemin est **simple** si chaque *arc* dans le chemin apparaît au plus une seule fois.
- Une chemin est dit **élémentaire** si chaque sommet dans le chemin apparaît une seule fois.
- Un **circuit** est un chemin dont les sommets de départ et de fin sont les mêmes. C'est donc une suite de la forme

$$(v_0, e_1, v_1, \dots, e_k, v_0)$$

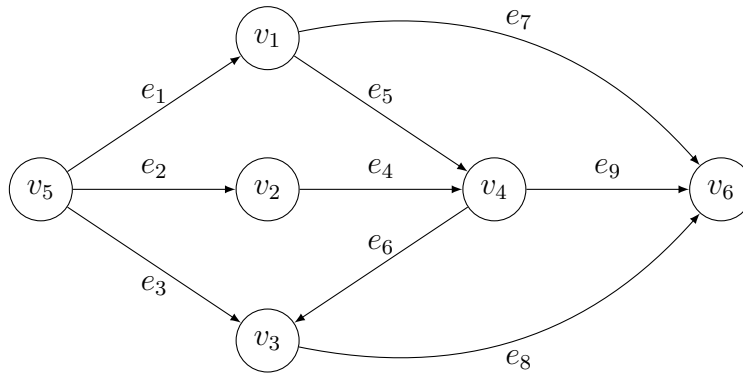
avec $e_j = (v_{j-1}, v_j)$ pour $j < k$, et $e_k = (v_{k-1}, v_0)$.

Exemple 29. *le graphe ci-dessous contient entre autres les chemins $(v_5, e_1, v_1, e_5, v_4, e_6, v_3, e_8, v_6)$ et $(v_4, e_9, v_6, e_{10}, v_4, e_6, v_3)$ de longueur respectives 4 et 3. $(v_5, e_2, v_2, e_4, v_4, e_6, v_3, e_3, v_5)$ est un circuit de longueur 4 de ce graphe.*



Définition 2.5.1. Un graphe orienté qui ne possède pas de circuit est **acyclique**

Exemple 30. le graphe ci-dessous acyclique.



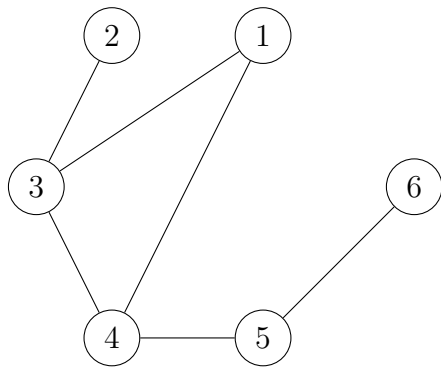
Remarque 2.5.1. Des définitions similaires peuvent être données pour les graphes non orientés.

2.6 Graphe partiel et sous graphe

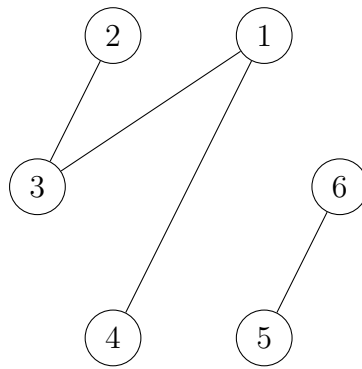
Définition 2.6.1. Soit $G = (V, E)$ un graphe un graphe non orienté. Le graphe $G' = (V, E')$ est un **graphe partiel** de G , si E' est inclus dans E . Autrement dit, on obtient un graphe partiel G' en enlevant une ou plusieurs arêtes au graphe G .

Définition 2.6.2. Soit S un sous-ensemble de sommets inclus dans V . On appelle sous-graphe de G induit par S , le graphe $G' = (S, E(S))$ dont l'ensemble des sommets est S et l'ensemble des arêtes $E(S)$ est formé de toutes les arêtes de G ayant leur deux extrémités dans S . Autrement dit, on obtient un sous-graphe G' de G en enlevant un ou plusieurs sommets au graphe G , ainsi que toutes les arêtes incidentes à ces sommets.

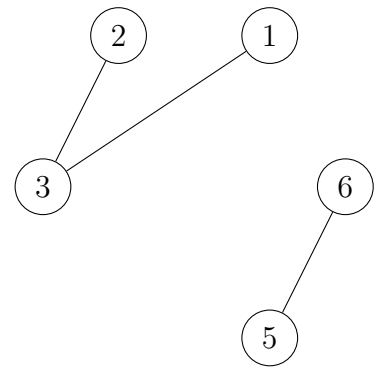
Exemple 31. Considérons le graphe $G = (V, E)$ suivant :



Graphe G



Graphe partiel de G



Sous-graphe de G

$$V = \{1, 2, 3, 4, 5, 6\}$$

$$E = \{\{1, 3\}, \{1, 4\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}\}$$

$$V = \{1, 2, 3, 4, 5, 6\}$$

$$E = \{\{1, 3\}, \{1, 4\}, \{2, 3\}, \{5, 6\}\}$$

$$V = \{1, 2, 3, 5, 6\}$$

$$E = \{\{1, 3\}, \{2, 3\}, \{5, 6\}\}$$

Définition 2.6.3. Une **clique** d'un graphe G est un sous-graphe complet de G .

Dans le graphe G de l'exemple ci-dessus, le sous-graphe $K = (V, E)$ avec $V = \{1, 3, 4\}$ et $E = \{\{1, 3\}, \{1, 4\}, \{3, 4\}\}$ est une clique.

Remarque 2.6.1. Cette notion s'applique aussi aux graphes orientés.

2.7 Coloration des sommets d'un graphe

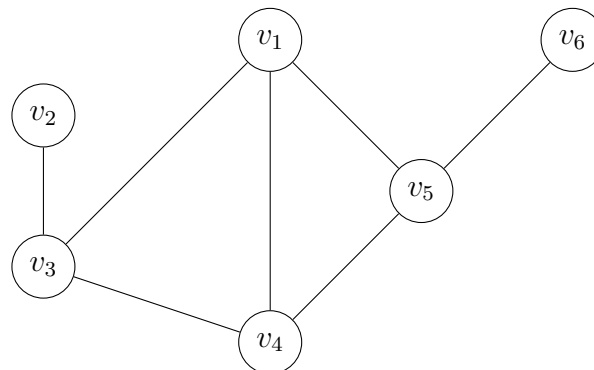
2.7.1 Stable d'un graphe

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

Un sous ensemble S de l'ensemble V des sommets du graphe G est un **stable** de G si S est un ensemble de sommets deux à deux non adjacents.

La cardinalité d'un stable est le nombre de sommets qu'il contient.

Exemple 32. Considérons le graphe $G = (V, E)$ suivant :



Dans ce graphe, les sous ensembles de V suivants sont des stables du graphe G :

$$S_1 = \{v_2, v_4, v_6\}, S_2 = \{v_1, v_2, v_6\}, S_3 = \{v_2, v_5\}, S_4 = \{v_2, v_4\} \dots$$

2.7.2 Définition de la coloration des sommets d'un graphe

Elle consiste à affecter à tous les sommets d'un graphe un entier naturel appelé **couleur** de telle sorte que deux sommets adjacents ne portent pas la même **couleur**. Mathématiquement, la coloration des sommets d'un graphe $G = (V, E)$ est donc une application $c : V \rightarrow \mathbb{N}^*$ (les entiers $\mathbb{N}$ sont appelés couleurs) telle que pour toute paire $\{u, v\}$ de sommets adjacents on ait $c(v) \neq c(u)$. Ainsi les sommets qui ont même couleur forment un stable du graphe. Autrement dit, un stable peut être identifié par une couleur. Une coloration des sommets d'un graphe avec k couleurs est donc une partition de l'ensemble des sommets du graphe en k stables.

La notion de coloration qui est une notion de la théorie des graphes est très utile en recherche opérationnelle. Une question que l'on peut se poser est que, étant donné un graphe, quel est le nombre minimum de couleurs possibles qu'il faut pour une coloration du graphe. Ce nombre minimum noté $\chi(G)$, s'appelle le **nombre chromatique** du graphe G . En d'autres termes, le nombre chromatique du graphe G est le plus petit entier k pour lequel il existe une partition de V en k stables.

Le nombre chromatique d'un graphe complet G est égal à l'ordre du graphe.

2.7.3 Encadrement du nombre chromatique

Proposition 1

Soit G un graphe non orienté. Le nombre chromatique $\chi(G)$ de G est inférieur ou égal à $\deg(G) + 1$:

$$\chi(G) \leq \deg(G) + 1$$

Proposition 2

Soit G un graphe non orienté simple. Le nombre chromatique $\chi(G)$ de G est supérieur ou égal au nombre chromatique de chacun de ses sous-graphes. En particulier, le nombre chromatique de G est supérieur ou égal à l'ordre $\omega(G)$ de sa plus grande clique (la plus grande clique est celle ayant le plus grand nombre de sommets).

$$\chi(G) \geq \omega(G).$$

Ainsi nous avons

$$\omega(G) \leq \chi(G) \leq \deg(G) + 1$$

Proposition 3

Soit $G = (V, E)$ un graphe non orienté simple. on a :

$$\left\lceil \frac{|V|}{\alpha(G)} \right\rceil \leq \chi(G) \leq |V| + 1 - \alpha(G)$$

où $\alpha(G)$ le cardinal maximal obtenu par un ensemble stable.

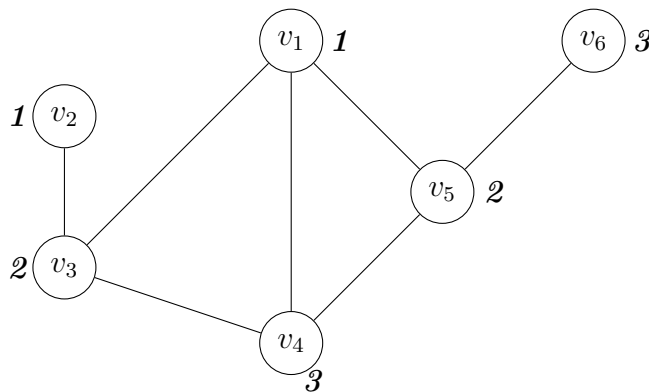
$\left\lceil \frac{|V|}{\alpha(G)} \right\rceil$ signifie partie entière de $\frac{|V|}{\alpha(G)}$.

Théorème 2.7.1. *Soit G est un graphe simple connexe. Si G n'est ni un graphe complet, ni un cycle de longueur impaire alors $\chi(G) \leq \deg(G)$.*

Corollaire 2.7.1. *Si G est un graphe simple connexe qui n'est pas régulier, alors*

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

Exemple 33. *Sur le graphe ci-dessous, on a eu besoin de trois couleurs (notées 1, 2 et 3) pour colorer les sommets de sorte que deux sommets adjacents aient des couleurs différentes. On a donc trois stables : $\{v_1, v_2\}$, $\{v_3, v_5\}$, $\{v_4, v_6\}$. On ne peut pas utiliser moins de couleurs, à cause des cliques $\{v_1, v_3, v_4\}$ et $\{v_1, v_4, v_5\}$.*



2.7.4 Algorithme glouton

L'algorithme glouton donne facilement un coloriage du graphe. le principe consiste à prendre les sommets les uns après les autres et d'affecter à chaque s choisi la couleur minimale (car une couleur est un entier naturel) qui n'apparaît pas dans les voisins coloriés de s .

Algorithme glouton de coloriage d'un graphe

Données : Un graphe $G = (V, E)$

Résultat : Une coloration $c : V \rightarrow \mathbb{N}^*$

Pour $s \in V$, faire

$c(s) \leftarrow$ plus petite couleur non utilisé par les voisins de s .

condition de fin de l'algorithme : L'algorithme prend fin dès que tous les sommets sont coloriés.

L'algorithme glouton assure une bonne coloration du graphe, mais qui n'est pas forcément optimale. L'optimalité de l'algorithme glouton dans la coloration du graphe dépend généralement de l'ordre dans lequel on choisit les sommets.

2.7.5 Algorithme de Welsh-Powell

Cette algorithme couramment utilisé 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 soit minimum.

Algorithme de Welsh-Powell

étape1 : ranger les sommets du graphe dans l'ordre décroissant de leur degré.

étape2 : En parcourant les sommets dans l'ordre décroissant, attribuer une couleur non encore utilisée au premier sommet non encore coloré, et attribuer cette même couleur à chaque sommet non encoré coloré et non adjacents à un sommet de cette couleur.

étape3 : s'il reste des sommets non colorés, revenir à l'étape 2. Sinon, FIN.

2.8 Représentation non graphique d'un graphe

2.8.1 Représentation non graphique d'un graphe non orienté

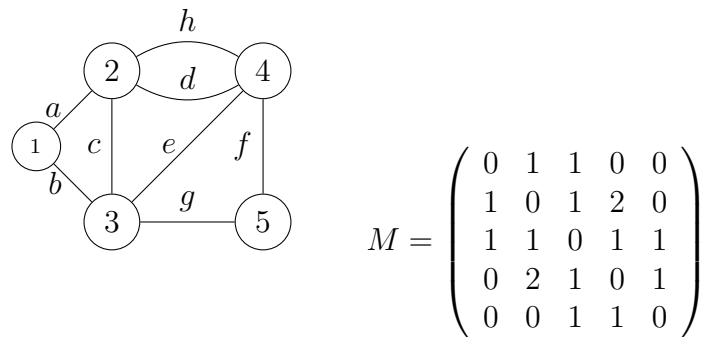
Matrice d'adjacences

Définition 2.8.1. Soit $G = (V, E)$ un graphe non orienté dont les sommets sont numérotés de 1 à n . La matrice d'adjacence de G est la matrice carrée $M = (m_{ij})$ définie par

$$m_{ij} = \begin{cases} k & \text{s'il y'a } k \text{ arêtes allant de } i \text{ à } j \\ 0 & \text{sinon} \end{cases}$$

Dans le cas non orienté cette matrice est symétrique.

Exemple 34. La matrice d'adjacence M du graphe ci-dessous est donnée ci-contre.

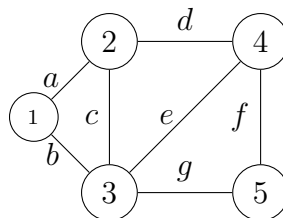


Théorème 2.8.1. Soit M^p la puissance p -ième d'une matrice d'adjacence M d'un graphe G . Le coefficient M^p_{ij} de l'élément de la i -ème ligne et de la j -ème colonne de M^p est égal au nombre de chemins de longueur p du graphe G allant du sommet i au sommet j .

Listes d'adjacences

On peut aussi représenter un graphe non orienté en donnant pour chacun de ses sommets la liste des sommets auxquels il est adjacent : ce sont **les listes d'adjacences**.

Exemple 35. Listes d'adjacences.



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

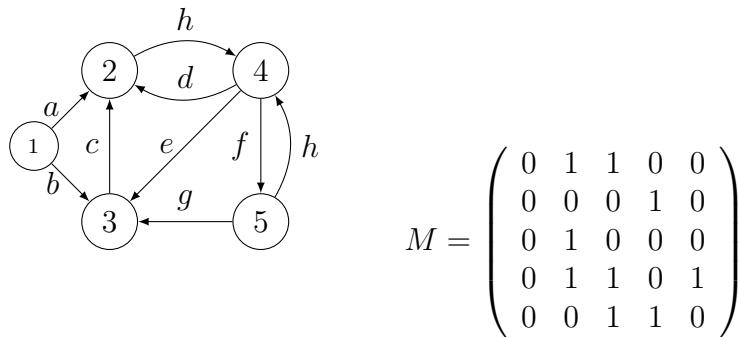
2.8.2 Représentation non graphique de digraphe

Matrice d'adjacences

Définition 2.8.2. Soit $G = (V, E)$ un graphe orienté dont les sommets sont numérotés de 1 à n . La matrice d'adjacence de G est la matrice carrée $M = (m_{ij})$ définie par

$$m_{ij} = \begin{cases} k & \text{s'il y'a } k \text{ arcs allant de } i \text{ à } j \\ 0 & \text{sinon} \end{cases}$$

Exemple 36. La matrice M d'adjacence du digraphe ci-dessous est donnée ci-contre.



Théorème 2.8.2. Soit M^p la puissance p -ième d'une matrice d'adjacence M d'un graphe G . Le coefficient M_{ij}^p de la i -ème ligne et de la j -ème colonne de M^p est égal au nombre de chemins de longueur p du graphe G dont l'origine est le sommet i et l'extrémité est le sommet j .

Matrice d'incidence (sommet-arc)

Un graphe peut être représenté par une matrice $n \times m$ ($n = |V|$ et $m = |E|$), dite d'incidence, pouvant contenir uniquement les valeurs 0, 1, -1. Chaque ligne de la matrice est associée à un sommet et chaque colonne à un arc. Ainsi, la valeur sur la i -ème ligne, j -ème colonne indique la relation qui existe entre un sommet et un arc.

1. 0 signifie que le sommet et l'arc ne sont pas adjacents,
2. 1 signifie que le sommet est l'extrémité initiale de l'arc,
3. -1 signifie que le sommet est l'extrémité terminale de l'arc.

Exemple 37. La matrice d'incidence du digraphe précédent est donnée par

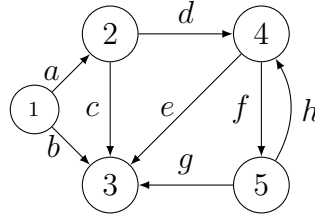
$$M = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{pmatrix}$$

L'ordre des lignes est pris suivant la numérotation des sommets de ce digraphe. L'ordre des colonnes est pris suivant l'ordre alphabétique attribué aux arcs.

Listes d'adjacences

On peut aussi représenter un digraphe en donnant pour chacun de ses sommets la liste des sommets qu'on peut atteindre directement en suivant un arc (dans le sens de la flèche).

Exemple 38. *Listes d'adjacences.*



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

2.9 Détermination des plus courts chemins

Définition 2.9.1. Soit $G = (E, A)$ un graphe orienté. On peut affecter à chaque arc (x, y) (x et y sont des sommets) de G un nombre réel $\omega(x, y)$ appelé le poids de l'arc (x, y) . Autrement dit on définit sur l'ensemble A des arcs de G , application ω appelée poids qui à chaque (x, y) associe un nombre réel :

$$\begin{aligned} \omega : \quad A &\rightarrow \mathbb{R} \\ (x, y) &\mapsto \omega(x, y) \end{aligned}$$

Définition 2.9.2. Soit G un graphe orienté muni d'un poids ω . En théorie des graphes, le problème de plus court chemin est un problème algorithmique qui consiste à trouver le plus court chemin qui va d'un sommet s de G à un sommet t donné de G .

La longueur d'un chemin est la somme des poids des arcs de ce chemin. Ainsi l'algorithme du plus court chemin qui va d'un sommet s à un sommet t consiste à trouver le chemin de s à t dont la somme des poids des arcs est la plus petite.

Les problèmes des plus courts chemins apparaissent naturellement dans des contextes variés. Ils peuvent apparaître comme modélisation de certains problèmes de recherche opérationnelle (trajet le plus rapide, ou le moins cher, gestion optimal d'un stock etc.). Ils sont aussi fréquemment des sous-problèmes d'autres problèmes d'optimisation (flot de coût minimum ou en théorie de l'ordonnancement). Les problèmes de plus court chemin peuvent être modélisés par les graphes orientés ou non et les arcs ou les arêtes peuvent avoir des poids positifs ou quelconques,... Nous allons nous intéresser ici à résolution des problèmes de plus courts chemins en utilisant des graphes orientés acycliques avec des poids positifs.

Algorithme de Dijkstra

C'est un algorithme qui permet de résoudre des problèmes de plus courts chemins en utilisant des graphes orientés acycliques avec des poids strictement positifs sur les arcs. **En 1959, Dijkstra** proposa le premier algorithme efficace pour résoudre le problème de plus court chemin. On suppose

que l'on a un graphe orienté acyclique $G = (V, A)$, avec une fonction de poids $\omega : A \rightarrow \mathbb{R}_+$. On se fixe un sommet s de départ. on suppose qu'à partir de s on peut rejoindre tous les autres sommets du graphe en suivant des arcs. Dans ce cas l'algorithme de Dijkstra permet de déterminer le plus court chemin de s à un sommet quelconque u du graphe. En particulier il permet de déterminer le plus court chemin de s à un sommet t donné.

Au cours de l'algorithme, on maintient un ensemble $U \subseteq V$ et une fonction $\lambda : V \rightarrow \mathbb{R}_+$ qui à chaque sommet $v \in V$ associe $\lambda(v)$ qui est égal à la longueur du plus court chemin qui va de s à v . Il est évident que $\lambda(s) = 0$.

Algorithme

1. Initialement on a :

$U \leftarrow V$;

Pour tout sommet $v \neq s$, faire $\lambda(v) = +\infty$ et $\forall v \in V$, faire $pred(v) = NULL$.

où $pred(v)$ indique le prédécesseur de v sur le plus court chemin de s à v .

2. Ensuite on répète la chose suivante de manière itérative :

Tant que $U \neq \emptyset$, répéter les instructions :

(a) exhiber l'élément u de U qui minimise l'ensemble $\lambda(U) = \{\lambda(x), x \in U\}$;

$U \leftarrow U \setminus \{u\}$;

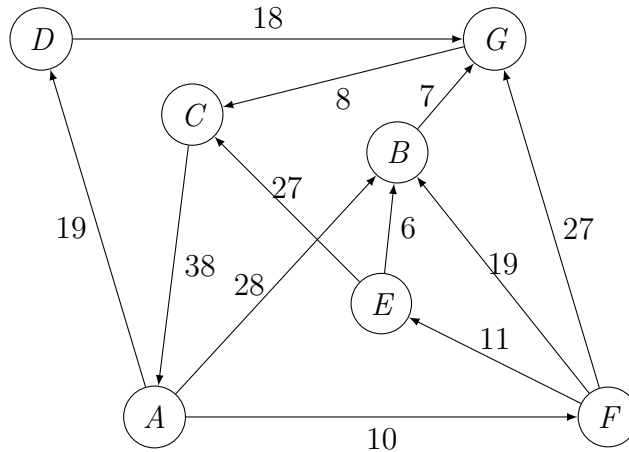
(b) **Pour chaque arc $(u, v) \in \delta^+(u)$, si $\lambda(v) > \lambda(u) + w(u, v)$, faire**

$\lambda(v) \leftarrow \lambda(u) + \omega(u, v)$;

$pred(v) := u$;

On s'arrête lorsque U est vide.

Exemple 39. Le graphe ci-dessous indique les différentes liaisons entre plusieurs lieux. le long de chaque arc figure la distance en kilomètres permettant d'aller d'un lieu à un autre selon le sens de la flèche. Un automobiliste doit se rendre de A à C . En utilisant l'algorithme de Dijkstra, déterminer le chemin le plus court que doit emprunter l'automobiliste.



Appliquons l'algorithme de Dijkstra.

itération 0 :initialisation

$U \leftarrow \{A, B, C, D, E, F, G\};$
 $\lambda(U) = \{\lambda(A), \lambda(B), \lambda(C), \lambda(D), \lambda(E), \lambda(F), \lambda(G)\} = \{0, \infty, \infty, \infty, \infty, \infty, \infty\};$
 $\forall v \in U, \text{pred}(v) = \text{null};$

itération 1 :

$U = \{A, B, C, D, E, F, G\} \neq \emptyset;$
 A *minimise* $\lambda(U) = \{\lambda(A), \lambda(B), \lambda(C), \lambda(D), \lambda(E), \lambda(F), \lambda(G)\} = \{0, \infty, \infty, \infty, \infty, \infty, \infty\};$
 $U \leftarrow U \setminus \{A\} = \{B, C, D, E, F, G\};$
 $\delta^+(A) = \{(A, B), (A, F), (A, D)\}$

* **Pour l'arc** $(A, B) \in \delta^+(A)$, **on a** : $\lambda(B) = \infty$ **et** $\lambda(A) + \omega(A, B) = 0 + 28 = 28;$
 $\lambda(B) > \lambda(A) + \omega(A, B);$ **donc** $\lambda(B) \leftarrow 28$
 $\text{pred}(B) = A;$

* **Pour l'arc** $(A, F) \in \delta^+(A)$, **on a** : $\lambda(F) = \infty$ **et** $\lambda(A) + \omega(A, F) = 0 + 10 = 10;$
 $\lambda(F) > \lambda(A) + \omega(A, F);$ **donc** $\lambda(F) \leftarrow 10$
 $\text{pred}(F) = A;$

* **Pour l'arc** $(A, D) \in \delta^+(A)$, **on a** : $\lambda(D) = \infty$ **et** $\lambda(A) + \omega(A, D) = 0 + 19 = 19;$
 $\lambda(D) > \lambda(A) + \omega(A, D);$ **donc** $\lambda(D) \leftarrow 19$
 $\text{pred}(D) = A;$

itération 2 :

$$U = \{B, C, D, E, F, G\};$$

$$F \text{ minimise } \lambda(U) = \{\lambda(B), \lambda(C), \lambda(D), \lambda(E), \lambda(F), \lambda(G)\} = \{28, \infty, 19, \infty, 10, \infty\};$$

$$U \leftarrow U \setminus \{F\} = \{B, C, D, E, G\};$$

$$\delta^+(F) = \{(F, B), (F, E), (F, G)\};$$

$$* \text{ Pour l'arc } (F, B) \in \delta^+(F), \text{ on a : } \lambda(B) = 28 \text{ et } \lambda(F) + \omega(F, B) = 10 + 19 = 29;$$

ici $\lambda(B) < \lambda(F) + \omega(F, B)$; donc on ne fait rien (autrment dit $\lambda(B)$ garde sa valeur précédente c'est - - dire 28 et $\text{pred}(B) = A$)

$$* \text{ Pour l'arc } (F, E) \in \delta^+(F), \text{ on a : } \lambda(E) = \infty \text{ et } \lambda(F) + \omega(F, E) = 10 + 11 = 21;$$

$$\lambda(E) > \lambda(F) + \omega(F, E); \text{ donc } \lambda(E) \leftarrow 21$$

$$\text{pred}(E) = F;$$

$$* \text{ Pour l'arc } (F, G) \in \delta^+(F), \text{ on a : } \lambda(G) = \infty \text{ et } \lambda(F) + \omega(F, G) = 10 + 27 = 37;$$

$$\lambda(G) > \lambda(F) + \omega(F, G); \text{ donc } \lambda(G) \leftarrow 37$$

$$\text{pred}(G) = F;$$

itération 3 :

$$U = \{B, C, D, E, G\};$$

$$D \text{ minimise } \lambda(U) = \{\lambda(B), \lambda(C), \lambda(D), \lambda(E), \lambda(G)\} = \{28, \infty, 19, 21, 37\};$$

$$U \leftarrow U \setminus \{D\} = \{B, C, E, G\};$$

$$\delta^+(D) = \{(D, G)\}; \lambda(G) = 37 \text{ et } \lambda(D) + \omega(D, G) = 19 + 18 = 37;$$

$$\lambda(G) = \lambda(F) + \omega(F, G); \text{ donc on ne fait rien.}$$

itération 4 :

$$U = \{B, C, E, G\};$$

$$E \text{ minimise } \lambda(U) = \{\lambda(B), \lambda(C), \lambda(E), \lambda(G)\} = \{28, \infty, 21, 37\};$$

$$U \leftarrow U \setminus \{E\} = \{B, C, G\};$$

$$\delta^+(E) = \{(E, B), (E, C)\};$$

$$* \text{ Pour l'arc } (E, B) \in \delta^+(E), \text{ on a : } \lambda(B) = 28 \text{ et } \lambda(E) + \omega(E, B) = 21 + 6 = 27;$$

$$\text{ici } \lambda(B) > \lambda(E) + \omega(E, B); \text{ donc } \lambda(B) \leftarrow 27$$

$$\text{pred}(B) = E;$$

$$* \text{ Pour l'arc } (E, C) \in \delta^+(E), \text{ on a : } \lambda(C) = \infty \text{ et } \lambda(E) + \omega(E, C) = 21 + 27 = 48;$$

$$\lambda(C) > \lambda(E) + \omega(E, C); \text{ donc } \lambda(C) \leftarrow 48$$

$$\text{pred}(C) = E;$$

itération 5 :

$$U = \{B, C, G\};$$

$$B \text{ minimise } \lambda(U) = \{\lambda(B), \lambda(C), \lambda(G)\} = \{27, 48, 37\};$$

$$U \leftarrow U \setminus \{B\} = \{C, G\};$$

$$\delta^+(B) = \{(B, G)\};$$

$$\lambda(G) = 37 \text{ et } \lambda(B) + \omega(B, G) = 27 + 7 = 34;$$

$$\text{comme } \lambda(G) > \lambda(B) + \omega(B, G); \text{ donc } \lambda(G) \leftarrow 34$$

$$\text{pred}(G) = B;$$

itération 6 :

$$U = \{C, G\};$$

$$G \text{ minimise } \lambda(U) = \{\lambda(C), \lambda(G)\} = \{48, 34\};$$

$$U \leftarrow U \setminus \{G\} = \{C\};$$

$$\delta^+(G) = \{(G, C)\};$$

$$\lambda(C) = 48 \text{ et } \lambda(G) + \omega(G, C) = 34 + 8 = 42;$$

$$\text{comme } \lambda(C) > \lambda(G) + \omega(G, C); \text{ donc } \lambda(C) \leftarrow 42$$

$$\text{pred}(C) = G;$$

itération 7 :

$$U = \{C\};$$

$$C \text{ minimise } \lambda(U) = \{\lambda(C)\} = \{48, 34\};$$

$$U \leftarrow U \setminus \{C\} = \{\};$$

$$\delta^+(C) = \{(C, A)\};$$

$$\lambda(A) = 0 \text{ et } \lambda(C) + \omega(C, A) = 42 + 38 = 80;$$

comme $\lambda(A) < \lambda(C) + \omega(C, A)$; donc on ne fait rien.

Fin de l'algorithme car $U = \{\}$.

Les résultats de chaque itération sont stockés dans le tableau ci-dessous au fur et à mesure que l'algorithme de Dijkstra évolue.

itération	A	B	C	D	E	F	G	ensemble U à l'issue de l'itération
0	0	∞	∞	∞	∞	∞	∞	A, B, C, D, E, F, G
1	0	(28, A)	∞	(19, A)	∞	(10, A)	∞	B, C, D, E, F, G
2	0	(28, A)	∞	(19, A)	(21, F)	(10, A)	(37, F)	B, C, D, E, G
3	0	(28, A)	∞	(19, A)	(21, F)	(10, A)	(37, F)	B, C, E, G
4	0	(27, E)	(48, E)	(19, A)	(21, F)	(10, A)	(37, F)	B, C, G
5	0	(27, E)	(48, G)	(19, A)	(21, F)	(10, A)	(34, B)	C, G
6	0	(27, E)	(42, G)	(19, A)	(21, F)	(10, A)	(34, B)	C
7	0	(27, E)	(42, G)	(19, A)	(21, F)	(10, A)	(34, B)	$\emptyset$

Dans ce travail nous avons cherché le plus court chemin qui mène de A à C. Mais en fait l'algorithme de Dijkstra nous donne le plus court chemin qui part de A à tous les sommets du graphe comme nous l'avons précisé plus haut.

À chaque itération, le couple (Y, Z) dans la colonne d'un sommet X a la signification suivante : Y représente provisoirement la longueur du plus court chemin qui mène de A à X et Z représente le prédécesseur de X. Ainsi à la fin de l'algorithme, la dernière itération donne la longueur du plus court chemin qui mène de A à n'importe quel sommet X. Pour déterminer le plus court chemin de A à un sommet X, il suffit de remonter la liste des prédécesseurs en partant du sommet X. Dans ce tableau, à la dernière itération, le couple (42, G) est mentionné dans la colonne de C. Ainsi la longueur du plus court chemin de A à C est 42 km et le prédécesseur de C est le sommet G. Le prédécesseur de G est B, celui de B est E, celui de E est F et celui de F est A. Le trajet le plus court pour aller de A à C est alors A – F – E – B – G – C.

CHAPITRE 3

FLOT DANS LES GRAPHS

Introduction

La notion de flot dans un graphe est naturelle : étant donné un réseau de transport (train, tuyau, câbles électriques), avec des arcs de capacités différentes, on se demande quelle est la quantité maximale de biens qu'on peut faire transiter à travers ce réseau sans "l'endommager". L'objectif du problème de flot dans les graphes est donc de déterminer la quantité maximale de flot ou de flux d'un bien qui peut circuler le long d'un réseau de transport.

3.1 Définitions

Définition 3.1.1. Un réseau de transport $\mathcal{R}$ est un graphe orienté $G = (V, A)$ auquel on associe une application $c : A \rightarrow \mathbb{R}_+$ qui à chaque arc (i, j) de G fait correspondre un nombre positif $c(i, j)$ appelé la capacité de l'arc (i, j) . Dans un réseau de transport il existe :

1. un seul sommet ou nœud s qui n'a pas de prédécesseurs, tous les autres sommets possèdent au moins un prédécesseur. Ce sommet est appelé **l'entrée ou la source du réseau**.
2. également un seul sommet t qui n'a pas de successeurs ; tous les autres sommets possèdent au moins un successeur. Ce sommet est appelé la sortie ou le puits du réseau.

Le réseau $\mathcal{R}$ est alors noté $\mathcal{R} = (G = (V, A), s, t, c)$.

Définition 3.1.2. Soit $\mathcal{R} = (G = (V, A), s, t, c)$ un réseau de transport muni de sa capacité $c : A \rightarrow \mathbb{R}_+$, de sa source s et de son puits t .

Un flot dans le réseau $\mathcal{R}$ ou un flot circulant de s vers t est une application $f : A \rightarrow \mathbb{R}_+$ qui à chaque arc (i, j) du réseau associe un réel positif $f(i, j)$ tel que :

- * $f(i, j) \leq c(i, j)$ pour tout arc $(i, j) \in A$;
- * pour tout sommet j différent de s et t ,
$$\sum_{(i,j) \in \delta^-(j)} f(i, j) = \sum_{(j,k) \in \delta^+(j)} f(j, k).$$

interprétation des propriétés de la définition

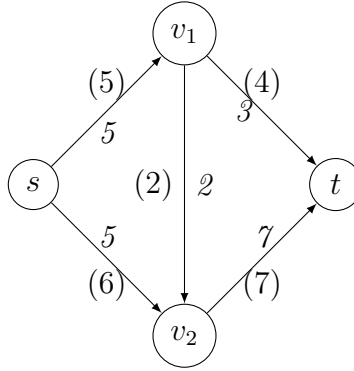
- Pour $(i, j) \in A$, $f(i, j)$ est le flot ou la quantité de flots ou encore le nombre d'unités circulant dans l'arc (i, j) .

- $f(i, j) \leq c(i, j)$ signifie que la quantité de flots circulant dans l'arc (i, j) est inférieure ou égale à la capacité $c(i, j)$ de l'arc (i, j) . Cette inégalité est appelée **la loi des capacités**.
- Pour un sommet j fixé dans $V \setminus \{s, t\}$, l'égalité $\sum_{(i,j) \in \delta^-(j)} f(i, j) = \sum_{(j,k) \in \delta^+(j)} f(j, k)$, appelée loi de conservation, signifie que la somme des flots entrant dans le sommet j est égale à la somme des flots sortant du même sommet j . En d'autres termes, la quantité totale de flots qui entre dans un sommet du réseau autre que s et t est égale à la quantité totale de flots qui sort de ce sommet. Cette égalité est appelée **la loi de conservation des sommets**.

Exemple 40. Dans ce graphe, les contraintes de capacités et les contraintes de conservation de flot sont vérifiées. Les nombres entre parenthèses sont les capacités. Ceux qui sont sans parenthèses représentent les flots.

En v_1 : 5, le flot entrant, est égale à $2 + 3$, où 2 et 3 sont les flots sortant.

En v_2 : $5 + 2$, le flot entrant, est égale à 7, le flot sortant.



Définition 3.1.3. Soit $\mathcal{R} = (G = (V, A), s, t, c)$ un réseau de transport muni de sa capacité $c : A \rightarrow \mathbb{R}_+$, de sa source s et de son puits t .

La valeur d'un flot f de s à t dans le réseau est la quantité de flot qui part de la source s . Elle est égale aussi à la quantité de flot qui arrive au puits t . On la note $\text{valeur}(f)$.

$$\text{valeur}(f) = \sum_{(s,i) \in \delta^+(s)} f(s, i) = \sum_{(k,t) \in \delta^-(t)} f(k, t).$$

L'objectif du problème de flot dans un réseau est de maximiser la valeur du flot dans le réseau.

Exemple 41. Dans l'exemple précédent, on a :

$$\text{valeur}(f) = f(s, v_1) + f(s, v_2) = 5 + 5 = f(v_1, t) + f(v_2, t) = 3 + 7 = 10$$

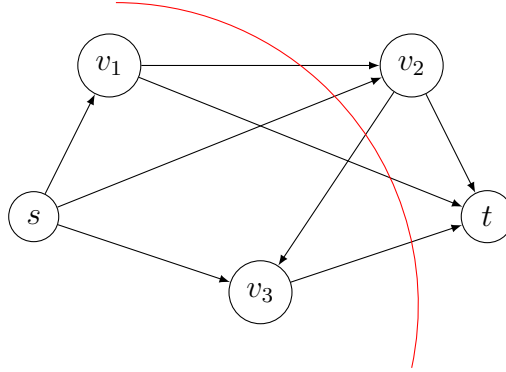
Définition 3.1.4. Un arc (i, j) est saturé si le flot dans cet arc est égal à la capacité de l'arc c'est-à-dire si $f(i, j) = c(i, j)$.

Exemple 42. Dans l'exemple précédent, les arcs (s, v_1) , (v_1, v_2) et (v_2, t) sont saturés.

Définition 3.1.5 (Coupe). Soit $\mathcal{R} = (G = (V, A), s, t, c)$ un réseau de transport muni de sa capacité $c : A \rightarrow \mathbb{R}_+$, de sa source s et de son puits t .

Une (s, t) -coupe ou une coupe du réseau $\mathcal{R}$ est une partition des sommets du réseau en deux sous ensembles $\mathcal{S}$ et $\mathcal{T}$ tels que $s \in \mathcal{S}$ et $t \in \mathcal{T}$. Elle est notée $(\mathcal{S}, \mathcal{T})$.

Exemple 43. Soit $\mathcal{S} = \{s, v_1, v_3\}$ et $\mathcal{T} = \{v_2, t\}$. Alors le couple $(\mathcal{S}, \mathcal{T})$ est une coupe du réseau suivant.



Définition 3.1.6 (capacité d'une coupe). Soit $G = (V, A)$ un graphe muni d'une capacité $c : A \rightarrow \mathbb{R}^+$. Soit $(\mathcal{S}, \mathcal{T})$ une (s, t) -coupe du réseau $\mathcal{R} = (G = (V, A), s, t, c)$. La capacité de la coupe $(\mathcal{S}, \mathcal{T})$, est la somme des capacités c_{ij} des arcs (i, j) du réseau tel que $i \in \mathcal{S}$ et $j \in \mathcal{T}$. Elle est notée $C(\mathcal{S}, \mathcal{T})$.

$$C(\mathcal{S}, \mathcal{T}) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{T}} c_{ij}.$$

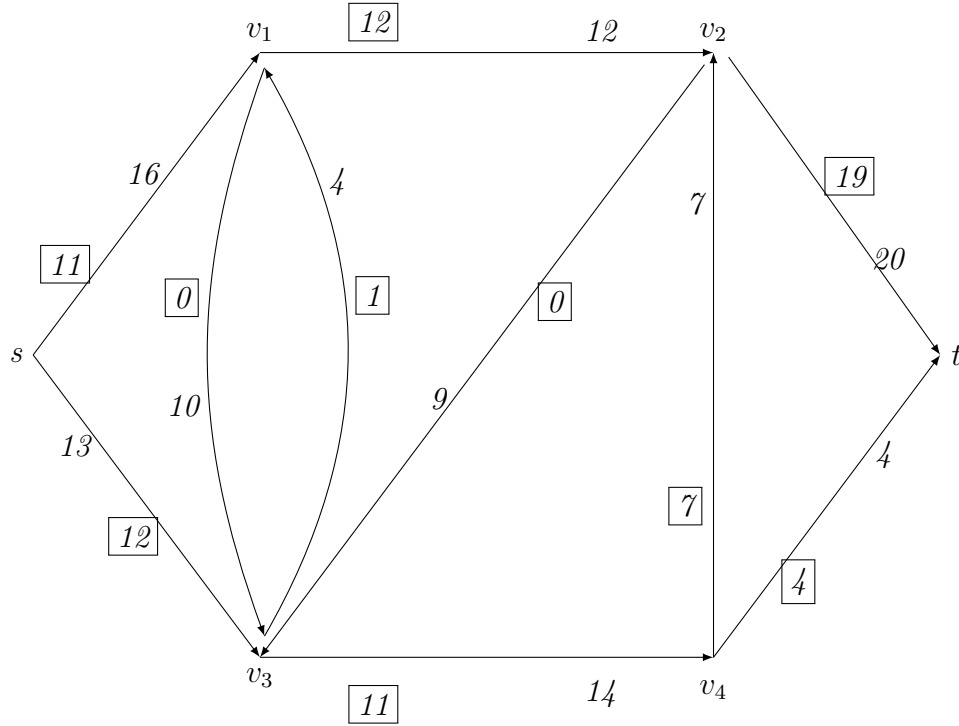
Théorème 3.1.1. Soit $D = (V, A)$ un réseau de transport muni de la capacité $c : A \rightarrow \mathbb{R}_+$, de sa source s et son puits t . Si f est une $s - t$ flot de valeur $\text{Valeur}(f)$ et si $(\mathcal{S}, \mathcal{T})$ est une (s, t) -coupe du réseau, alors

$$\text{Valeur}(f) \leq C(\mathcal{S}, \mathcal{T}).$$

Théorème 3.1.2 (Théorème du flot max-coupe min). La valeur maximum d'une $s - t$ flot est égale à la capacité minimum d'une (s, t) -coupe $(\mathcal{S}, \mathcal{T})$.

$$\max \text{Valeur}(f) = \min_{(\mathcal{S}, \mathcal{T})} C(\mathcal{S}, \mathcal{T}).$$

Exemple 44. On donne un réseau qui est le graphe représenté ci-dessous. Les nombres non encadrés sont les capacités des arcs. Ceux encadrés forment le flot. Préciser la valeur actuelle du flot dans le réseau. Est-elle maximum ?



3.2 Problème du flot maximum

Connaissant les capacités des arcs d'un réseau de transport, le problème du flot maximum consiste à trouver la quantité maximum de flot qui circule de la source s au puits t . Un des algorithmes le plus connu pour résoudre ce problème est celui de Ford et Fulkerson.

3.2.1 Algorithme de Ford et Fulkerson

Dans cette section nous développons l'algorithme de Ford-Fulkerson pour déterminer le flot maximum dans un réseau. Soit $\mathcal{R}$ un réseau de transport et f un flot dans le réseau $\mathcal{R}$. Dans le but de décrire l'algorithme de Ford-Fulkerson, nous allons d'abord définir le graphe résiduel associé au réseau $\mathcal{R} = (G = (V, A), s, t, c)$ muni du flot f .

Définition 3.2.1 (Graphe résiduel ou graphe d'écart-chemin f -augmentant). *Le graphe résiduel $\mathcal{R}_f$ associé au réseau $(\mathcal{R}, f)$ est un graphe orienté dont l'ensemble des sommets est exactement l'ensemble V des sommets du réseau $\mathcal{R}$ et dont l'ensemble A_f des arcs est construit de la manière suivante :*

Pour tout arc $(i, j) \in A$:

- si $f(i, j) < c(i, j)$, alors (i, j) est un arc du graphe résiduel avec pour capacité

$$c_f(i, j) = c(i, j) - f(i, j);$$

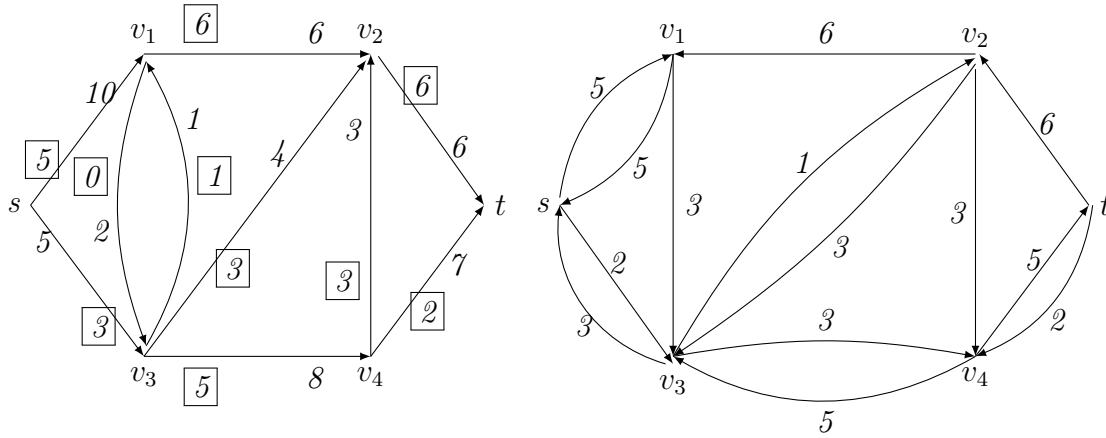
- si $f(i, j) > 0$, alors (j, i) est un arc du graphe résiduel avec pour capacité

$$c_f(j, i) = f(i, j).$$

L'avantage du graphe résiduel $\mathcal{R}_f = (V, A_f)$ est que tout chemin P de s à t dans $\mathcal{R}_f$ est un chemin le long duquel on peut augmenter ou diminuer la quantité de flot sans violer la loi de conservation des bornes de capacités.

- Tout chemin de s à t dans le graphe résiduel $\mathcal{R}_f$ est appelé **chemin f -augmentant** ou **chemin améliorant**.

Exemple 45. On donne un réseau qui est le graphe représenté à gauche. Les nombres non entourés sont les capacités des arcs. Ceux encadrés forment le flot. Le graphe résiduel de ce réseau est celui à droite.



Algorithme de Ford-Fulkerson 1956

On initialise le flot : poser $f(a) := 0$ pour tout $a \in A$. Ensuite on répète :

étape1 : Trouver un chemin f -augmentant P dans le graphe résiduel $\mathcal{R}_f$.

s'il n'y en a pas, alors le flot f dans le graphe G est optimal.

Sinon, on pose

$$\mu := \min_{(i,j) \in P} c_f(i,j) := \text{minimum des capacités résiduelles le long du chemin } f\text{-augmentant } P$$

et on va à l'étape 2

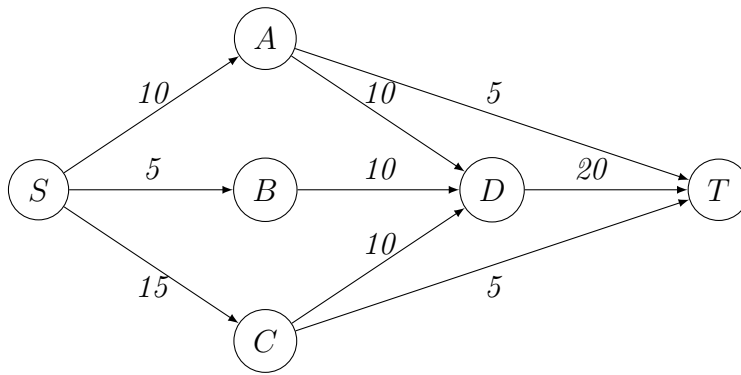
étape2 : On augmente le flot f dans le réseau $\mathcal{R}$ le long de P . Augmenter f le long de P se fait de la manière suivante :

$$\begin{cases} f(i,j) := f(i,j) + \mu & \text{si } (i,j) \in P \text{ est dans le même sens dans } A \text{ (i.e } (i,j) \in A) \\ f(j,i) := f(j,i) - \mu & \text{si } (i,j) \in P \text{ et } (j,i) \in A \end{cases}$$

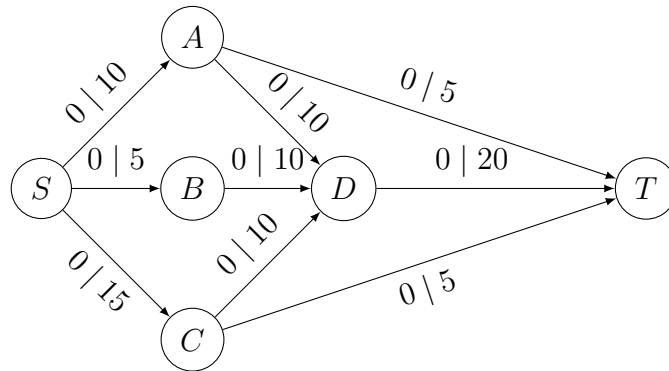
Aller à l'étape 1.

Dès qu'il n'y a plus de chemin f -augmentant dans le graphe résiduel, alors l'algorithme prend fin. Ainsi la valeur maximale du flot dans le réseau est la dernière valeur du flot dans le réseau.

Exemple 46. On considère le réseau suivant dans lequel chaque arc est muni de sa capacité. Déterminer à l'aide de l'algorithme de Ford-Fulkerson le flot maximal qui peut circuler dans ce réseau.

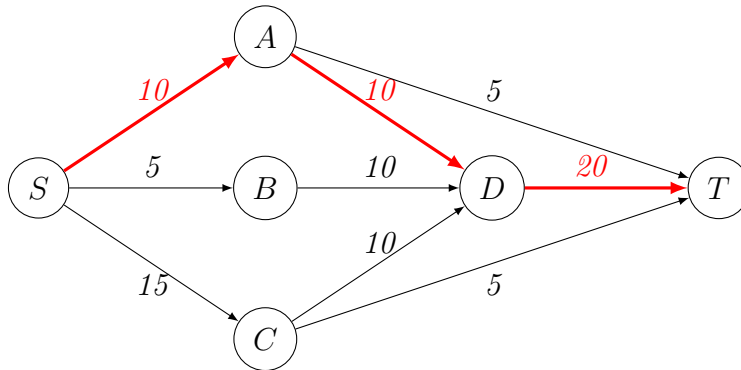


Résolution 1. D'après l'algorithme de Ford-Fulkerson, on initialise la valeur du flot dans le réseau égal à zéro : $\text{valeur}(f) \leftarrow 0$. Ainsi initialement le réseau se présente de la façon suivante :



Le flot étant la valeur se trouvant à gauche de la barre |.

Étape 0 : $GR(0) := \text{graphe résiduel initial}$

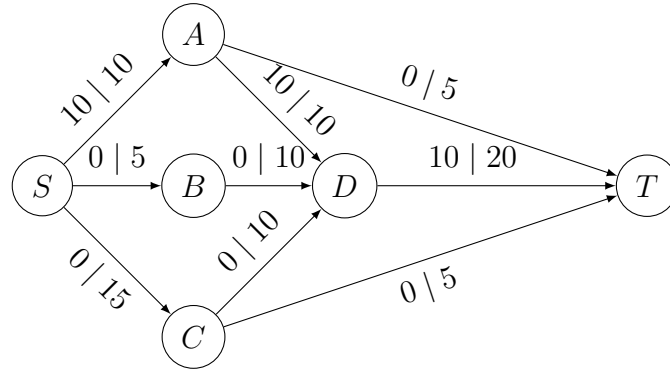


Dans ce graphe résiduel, il y'a plusieurs chemins améliorants :

1. SAT : le flot maximal qu'on peut augmenter le long de ce chemin est $\mu = \min \{10, 5\} = 5$;
2. SADT : le flot maximal qu'on peut augmenter le long de ce chemin est $\mu = \min \{10, 10, 20\} = 10$;
3. SBDT : le flot maximal qu'on peut augmenter le long de ce chemin est $\mu = \min \{5, 10, 20\} = 5$;
4. SCT : le flot maximal qu'on peut augmenter le long de ce chemin est $\mu = \min \{15, 5\} = 5$;
5. SCDT : le flot maximal qu'on peut augmenter le long de ce chemin est $\mu = \min \{15, 10, 20\} = 10$;

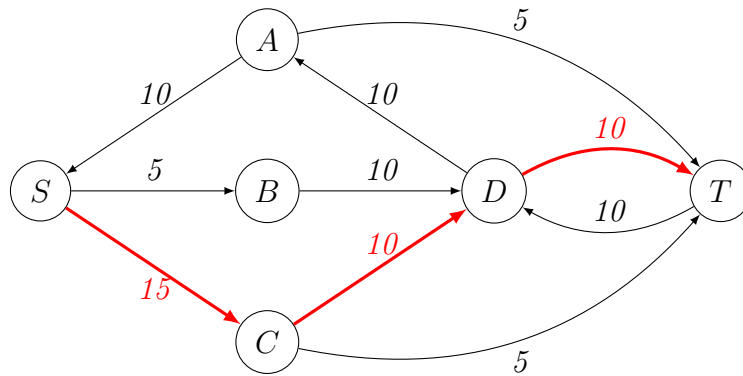
On peut choisir n'importe quel chemin améliorant pour augmenter la valeur du flot dans le réseau. Mais nous allons choisir un chemin améliorant qui permet d'augmenter le flot avec une grande valeur (ce choix nous permet d'atteindre rapidement le flot maximal). Nous allons donc choisir le chemin améliorant SADT. Ainsi on augmente le flot dans le réseau en augmentant le flot le long du chemin SADT.

$Valeur(flott) \leftarrow 0 + 10$. Ainsi l'état actuel du réseau est le suivant :



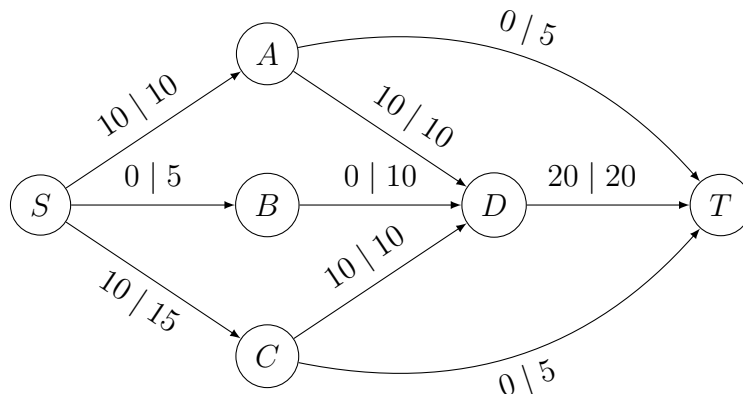
Étape 1 : GR(1).

Le graphe résiduel GR(1) correspondant à l'état actuel du réseau est donné par



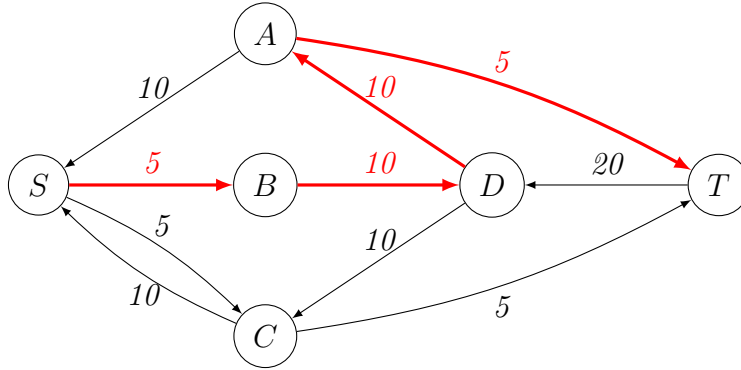
Dans ce graphe résiduel, il existe au moins un chemin améliorant qui est par exemple le chemin SBDT. Donc la valeur actuelle du flot dans le réseau n'est optimale. Ainsi on peut encore augmenter le flot dans le réseau. En considérant le chemin améliorant SCDT, on augmente le flot dans le réseau en augmentant le flot le long du chemin SCDT par la capacité résiduelle $\mu = \min \{15, 10, 10\} = 10$. Ainsi on obtient le réseau suivant avec la valeur du flot

$$valeur(f) \leftarrow 10 + 10 = 20.$$

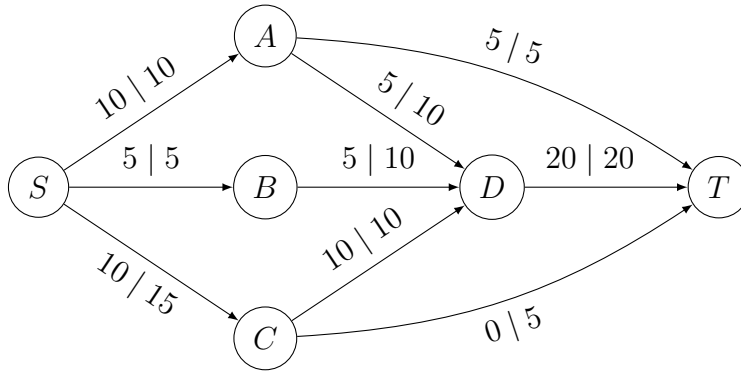


Étape 2 : $GR(2)$.

Le graphe résiduel $GR(2)$ correspondant à l'état actuel du réseau est donné par



$SBDAT$ est un chemin améliorant du réseau. On peut donc augmenter le flot dans le réseau en augmentant le flot le long du chemin $SBDAT$ par la capacité résiduelle $\mu = \min \{5, 10, 10, 5\} = 5$. Le réseau se présente alors comme suit :



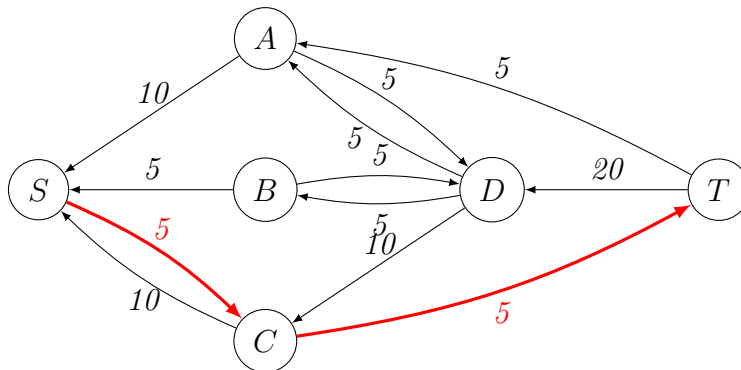
La valeur du flot dans le réseau est alors

$$\text{valeur}(f) \leftarrow 20 + 5 = 25.$$

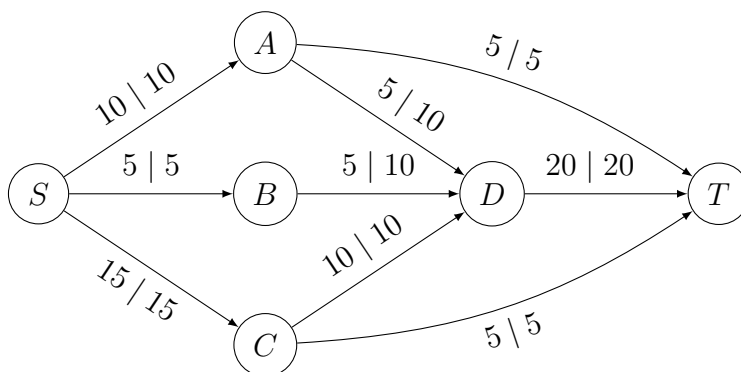
Remarque 3.2.1. Augmenter le flot de 5 dans l'arc (A,D) dans le réseau, revient à le diminuer car l'arc correspondant, dans le graphe résiduel qui est dans (D,A) , est dans le sens inverse de (A,D) .

Étape 3 : $GR(3)$.

Le graphe résiduel $GR(3)$ correspondant à l'état actuel du réseau est donné par



SCT est un chemin améliorant du réseau. On peut donc augmenter le flot dans le réseau en augmentant le flot le long du chemin SCT par la capacité résiduelle $\mu = \min \{5, 5\} = 5$. Le réseau de transport prend alors l'état suivant :

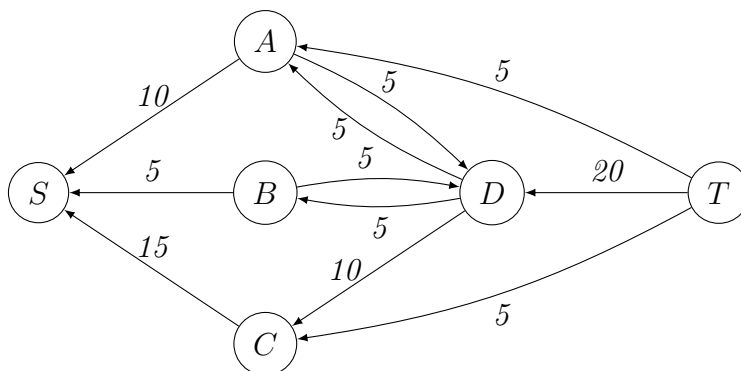


La valeur du flot dans le réseau est alors

$$\text{valeur}(f) \leftarrow 25 + 5 = 30.$$

Étape 4 : $GR(4)$.

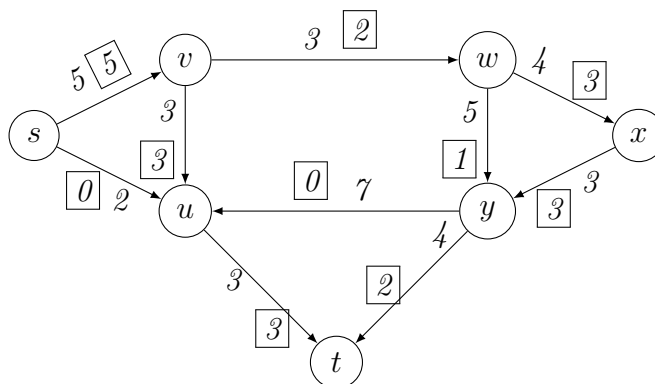
Le graphe résiduel $GR(4)$ correspondant à l'état actuel du réseau est donné par



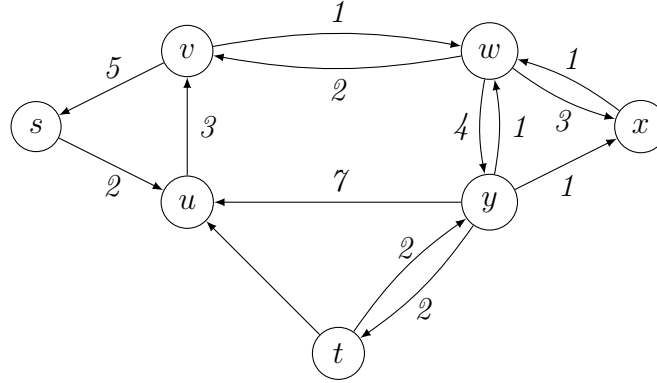
Il n'y a aucun chemin améliorant dans ce graphe résiduel. L'algorithme prend donc fin et la valeur maximale du flot dans le réseau est alors

$$\text{valeur}_{\max}(f) = 30.$$

Exemple 47. Voici un réseau de transport dans lequel circule un flot. Les nombres encadrés sont le flot, ceux non encadrés sont les capacités.



Le graphe d'écart correspondant est le suivant



$P = s - u - v - w - y - t$) est un chemin dans ce graphe d'écart pour aller de s à t . C'est donc un chemin f -augmentant. Le minimum des capacités résiduelles des arcs sur ce chemin P est $\mu = \min \{c_f(a) : a \in P\} = \min \{2, 3, 1, 4, 2\} = 1$.

* l'arc (s, u) de P est dans le même sens dans A , donc on augmente le flot de (s, u) de $\mu = 1$

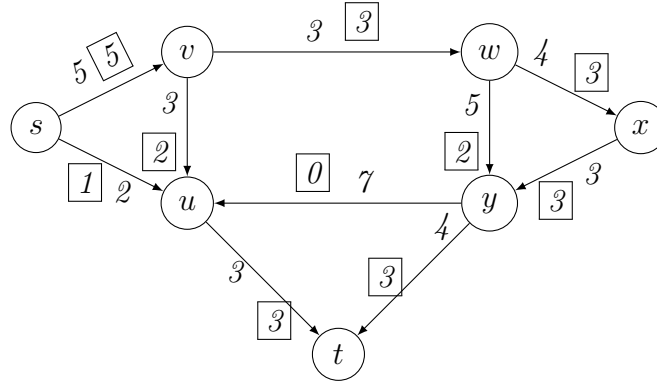
* l'arc (u, v) de P est dans le sens inverse que dans A , donc on diminue le flot (v, u) de $\mu = 1$

* l'arc (v, w) de P est dans le même sens dans A , donc on augmente le flot de (v, w) de $\mu = 1$

1. l'arc (w, y) de P est dans le même sens dans A , donc on augmente le flot de (w, y) de $\mu = 1$

2. l'arc (y, t) de P est dans le même sens dans A , donc on augmente le flot de (y, t) de $\mu = 1$

Ainsi on obtient le graphe,



Il n'y a plus de chemin f augmentant dans le graphe résiduel. Donc le flot actuel dans le réseau est le flot maximal.

TD de RO : Graphes et applications

Exercice 3.1. À partir de n intervalles $I_1, \dots, I_n$ de l'ensemble des réels $\mathbb{R}$ on peut construire un graphe G où les n sommets $1, \dots, n$ de G représentent respectivement les intervalles $I_1, \dots, I_n$. Dans un tel graphe, deux sommets i et j , $i \neq j$, sont adjacents si et seulement si $I_i \cap I_j \neq \emptyset$. Ce graphe est appelé graphe d'intervalles.

On considère les intervalles $I_1 =]-\infty; 0]$, $I_2 = [0; 1]$, $I_3 =]1; 3]$, $I_4 =]-1; \frac{1}{2}]$, $I_5 =] \frac{1}{2}; 5]$, $I_6 =]0; 5; 1[$, $I_7 =]9; +\infty[$.

1. Construire le graphe d'intervalles G correspondant à ces intervalles.

2. Quelle la nature de ce graphe.
3. Déterminer la matrice d'adjacence de ce graphe.

Exercice 3.2. Trois professeurs P_1, P_2, P_3 devront donner lundi prochain un certain nombre d'heures de cours à trois classes C_1, C_2, C_3 .

P_1 doit donner 2 heures de cours à C_1 et 1 heure de cours à C_2 ;

P_2 doit donner 1 heure de cours à C_1 , 1 heure de cours à C_2 et 1 heure de cours à C_3 ;

P_3 doit donner 1 heure de cours à C_1 , 1 heure de cours à C_2 et 2 heures de cours à C_3 .

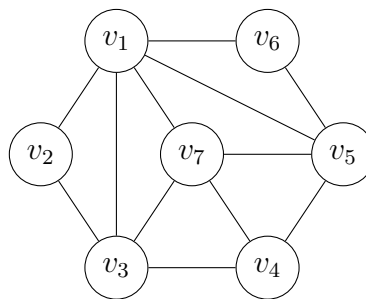
1. Représenter cette situation par un graphe.
2. Quel type de graphe obtenez-vous ?
3. Combien faudra-t-il de plages horaires au minimum ?
4. Aidez-vous du graphe pour proposer un horaire du lundi pour ces professeurs.

Indications : L'ensemble V des sommets est constitué des professeurs et des classes. Chaque arête correspond à **une heure**. Nous identifions une plage horaire à une couleur.

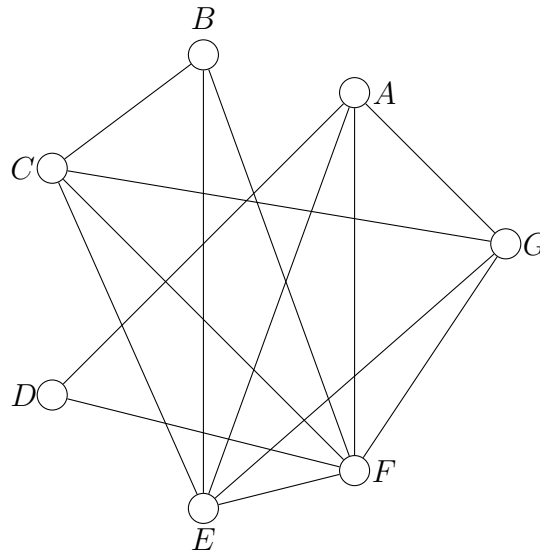
Exercice 3.3. Un tournoi d'échecs oppose 6 personnes. Chaque joueur doit affronter tous les autres.

1. Construisez un graphe représentant toutes les parties possibles.
2. Quel type de graphe obtenez-vous ?
3. Si chaque joueur ne joue qu'un match par jour, combien de jours faudra-t-il pour terminer le tournoi.
4. Aidez-vous du graphe pour proposer un calendrier des matches si chaque joueur ne joue qu'un match par jour.

Exercice 3.4. Donner un encadrement du nombre chromatique de ce graphe



Exercice 3.5. Un concert de solidarité est organisé dans une grande salle de spectacle. A ce concert sont conviés sept artistes de renommée internationale : Luther Allunison (A), John Biaise (B), Phil Colline (C), Bob Ditlane (D), Jimi Endisque (E), Robert Fripe (F) et Rory Garaguerre (G). Les différents musiciens invités refusant de jouer avec certains autres, l'organisateur du concert doit prévoir plusieurs parties de spectacle. Les arêtes du graphe ci-dessous indiquent quels sont les musiciens qui refusent de jouer entre eux.

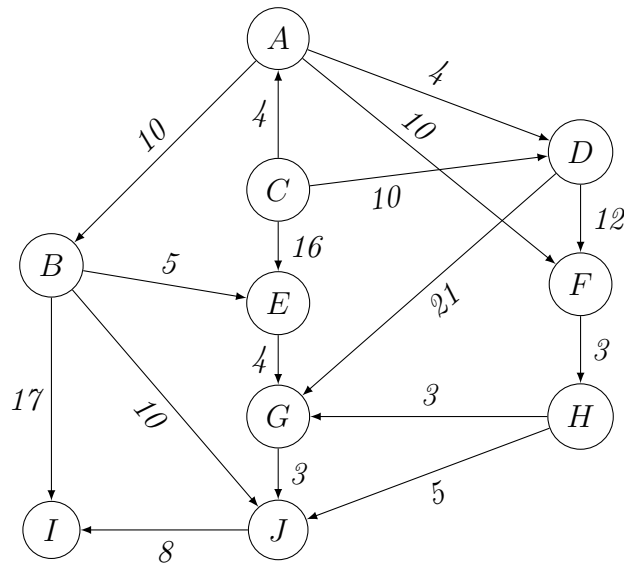


1. Quelle est la nature du graphe G .
2. Déterminer une clique de G ayant le plus grand ordre.
3. Déterminer un encadrement du nombre chromatique $\chi(G)$ de G .
4. En utilisant l'algorithme de Welsh-Powell, colorer le graphe G et préciser la valeur de $\chi(G)$.
5. Combien de concerts l'organisateur doit-il prévoir au minimum ? Proposer une répartition des musiciens pour chacun de ces concerts.

Exercice 3.6. Pour le bon fonctionnement de son entreprise, un directeur décide que ses employés Kpatcha, Joseph, Noe, Yvette, Elom et Wenpwire participent à une série de formations organisées en Comptabilité (C), Informatique (I), Droit (D), Anglais (A) et Marketing. La formation dans chaque domaine doit durer 1 heure et les employés n'ont pas le même niveau dans les domaines. Kpatcha doit suivre la comptabilité, l'informatique et le droit. Joseph ne doit suivre que la formation d'informatique. Noe doit suivre les formations de comptabilité et de droit. Yvette est appelée à suivre l'informatique, droit et anglais. Elom doit se faire former en anglais et marketing et Wenpwire en droit et marketing. Le but de l'exercice est de déterminer la durée minimale dans laquelle cette session de formation peut tenir pour que tous les employés soient formés comme prévu.

1. Construire le graphe G qui définit le plan de formation souhaité par le directeur.
2. Quelle type de graphe obtient-t-on ?
3. Dessiner un graphe G' dont les sommets sont les modules de formations et dont les arêtes décrivent les formations qui ne peuvent se produire au même moment.
4. Quelle la nature de ce graphe.
5. Déterminer alors la durée minimale de réalisation de la session de formation et donner si possible les formations qui peuvent se produire au même moment.

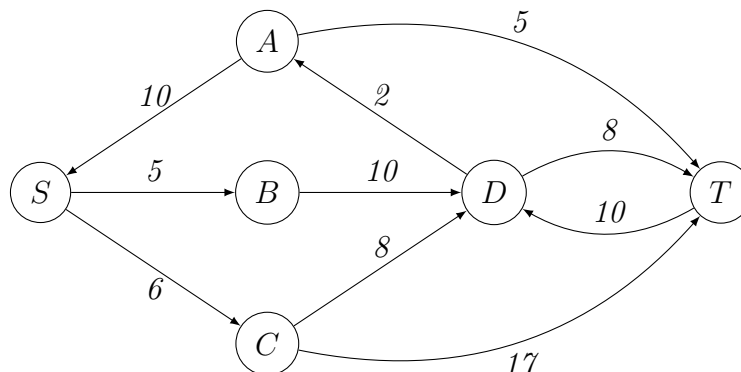
Exercice 3.7. On considère le graphe orienté suivant dont chaque arc porte un poids.



À l'aide de l'algorithme de Dijkstra, déterminer le plus court chemin qui mène de C à I et préciser sa longueur.

Exercice 3.8. Un livreur ayant quatre clients A, B, C, D prépare sa tournée. Il désire passer par certains de ses clients en partant de S pour arriver au point T. Les liaisons possibles sont représentées sur le graphe G ci-dessous dont chaque arc est muni d'un poids qui représente la durée, en minutes, qui sépare deux sommets.

1. À l'aide de l'algorithme de Dijkstra déterminer le chemin qu'il doit emprunter pour minimiser la durée totale du trajet de S à T? Préciser les clients qu'il pourra visiter dans ce cas.
2. Préciser le chemin qu'il va emprunter à son retour. Quelle est la durée de ce trajet?



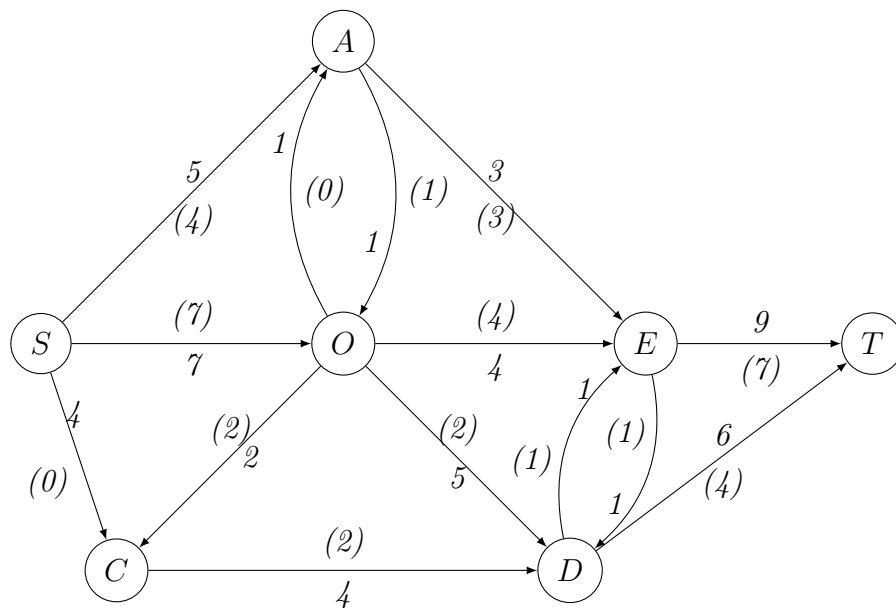
Exercice 3.9. .

A, B, C, D, E, F, G, H, et I sont les tâches à exécuter pour la réalisation d'un projet P. Dans le tableau ci-dessous, chaque tâche a été mentionnée avec sa durée d'exécution et la ou les tâche(s) antérieure(s) qui la précède(ent).

tâches	tâches antérieures	Durée (jours)
A	-	3
B	-	9
C	-	5
D	A	8
E	B	4
F	B	7
G	B	20
H	C, F	6
I	D, E	5

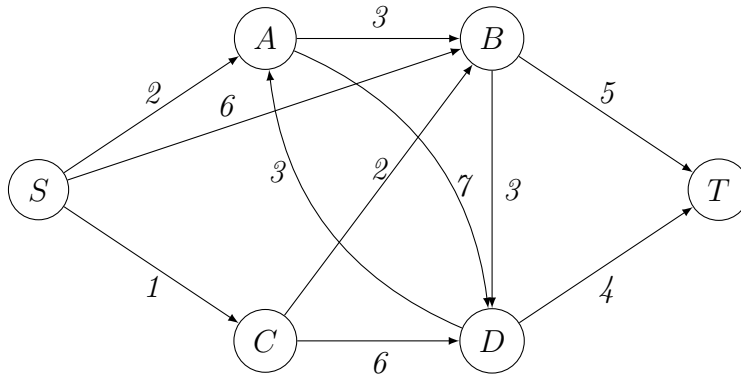
1. Proposer le graphe PERT qui modélise ce projet.
2. Déterminer les dates au plus tôt et les dates au plus tard de chacun des événements de début et de fin d'une tâche.
3. Préciser les tâches critiques de ce projet. Quelle est le chemin critique.
4. Déterminer la durée optimale de réalisation du projet.

Exercice 3.10. On considère le réseau suivant. Les nombres entre parenthèses représentent les flots. Ceux qui ne sont pas dans les parenthèses sont les capacités.



1. Déterminer le graphe résiduel correspondant à ce réseau.
2. Déterminer tous les chemins améliorant de ce réseau.

Exercice 3.11. Un serveur S est connecté à une machine T par un réseau avec les nœuds A , B , C et D . Les capacités de connexion entre les nœuds sont (en Mbit/s). Ce système informatique est représenté par le graphe orienté suivant :



L'utilisateur de la machine T télécharge un très grand fichier du serveur S . On veut déterminer la valeur du flot qui maximise le passage des données dans ce réseau.

1. Comment appelle-t-on ce problème qu'on veut résoudre ? Préciser l'algorithme du cours permettant de résoudre ce problème.
2. Appliquer cet algorithme pour déterminer le débit maximal de données qui doit circuler dans ce réseau.