

Mathématiques – Recherche Opérationnelle

Alain Billionnet
Notes de cours

2010–2011

Avertissement

Ce document ne prétend pas à l'exhaustivité d'un cours.

Il se peut que des erreurs ou des fautes de frappe s'y soient également glissées par mégarde, auquel cas je vous invite à envoyer vos corrections à l'adresse `marc.vanderwal@ensiie.fr`.

J'espère que ces notes vous seront néanmoins utiles pour vos révisions.

Marc 'xOr' van der Wal



Ce document est distribué sous la licence Creative Commons Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique version 2.0. Pour plus d'informations :

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

Table des matières

0	Introduction	9
	Généralités	9
	Types de problèmes étudiés	10
	Programmation linéaire	10
	Programmation linéaire en nombre entiers	10
	Aléatoire/Stochastique	10
1	Programmation dynamique	11
1.1	Généralités	11
1.2	Exemple : Tracé optimal d'une autoroute	11
1.3	Formalisation	12
1.3.1	Formules d'optimisation séquentielle	13
1.4	Deuxième exemple d'application	13
1.4.1	Représentation du problème en tant que graphe	15
2	Chemins optimaux dans les graphes	17
2.1	Position du problème	17
2.2	Algorithme de Dijkstra	18
2.2.1	Exemple d'application	19
2.2.2	Preuve de l'algorithme	19
2.3	Algorithme de Bellman–Ford	20
2.3.1	Exemple d'application	20
2.4	Algorithme de Floyd–Warshall	21
3	Flots dans les graphes	23
3.1	Généralités	23
3.2	Problème du flot maximal	25
3.2.1	Définitions	25
3.2.2	Condition nécessaire et suffisante d'optimalité	27
3.3	Algorithme de Ford–Fulkerson	27
3.3.1	L'algorithme	27

3.3.2	Exemple d'application	29
4	Ordonnement	31
4.1	Généralités	31
4.2	La méthode PERT	32
4.2.1	Exemple de diagramme	32
4.2.2	Exemple avec des tâches fictives	33
4.2.3	Définitions	33
4.2.4	Marges relatives aux tâches	35
4.2.5	Exemple d'ordonnement d'atelier	35
5	Méthodes d'énumération implicite	37
5.1	Idée sur un exemple	37
5.2	Schéma général	38
5.3	Application au problème du voyageur	40
5.3.1	Calcul du regret	41
6	Programmation linéaire	43
6.1	Problèmes de production	43
6.1.1	Optimisation fractionnaire	44
6.1.2	Interprétation géométrique d'un programme linéaire	45
6.1.3	Forme générale d'un programme linéaire	45
6.1.4	Base et solution de base	46
6.2	Algorithme du simplexe	48
6.2.1	Caractérisation des solutions optimales	48
6.2.2	Algorithme du simplexe pour la minimisation	52
6.2.3	Méthode des tableaux	52
6.3	Initialisation de l'algorithme du simplexe	54
6.3.1	Problème auxiliaire : Méthode des 2 phases	55
6.3.2	Exemples d'application	56
6.3.3	Méthode des pénalités	59
7	Processus aléatoires	61
7.1	Processus de Markov (sans mémoire)	61
7.1.1	Processus de Poisson	61
7.1.2	Processus de naissance	63
7.1.3	Processus de naissance et de mort	65
7.2	Chaînes de Markov	68
7.2.1	Matrice associée	68
7.2.2	Graphe associé à une chaîne	69
7.2.3	Classification des états	70

7.2.4	Distribution limite d'une chaîne de Markov finie	72
8	Phénomènes d'attente	75
8.1	Système à un guichet	75
8.1.1	Nombre moyen de clients dans le système	76
8.1.2	Nombre moyen de clients dans la file	77
8.2	Système à plusieurs guichets identiques	77
9	Fiabilité des équipements	79
9.1	Généralités	79
9.2	Formes analytiques de la fiabilité et du taux d'avarie	80
9.3	Généralisation de la loi exponentielle : loi Erlang-k	80
9.4	Matériel d'usure	81
9.5	Fiabilité des systèmes	81
9.5.1	Système en série	81
9.5.2	Système en parallèle	82
9.6	Probabilité de consommation	83
9.7	Entretien préventif	84
9.7.1	Calcul du coût moyen	84
9.7.2	Conclusion	85
A	Transformée de Laplace	87

Introduction

Généralités

Bibliographie

- Faure, Lemaire, Picouveau, *Précis de Recherche Opérationnelle*, Dunod.
- Roseaux, *Exercices et problèmes résolus de RO* (3 tomes), Dunod.

But du cours

- Savoir prendre les “meilleures” décisions (critère d’optimalité à définir)
- Démarche scientifique
- Aperçu des grands domaines de la RO

Historique

- Pascal, Fermat, Bernoulli
- Monge : problème des déblais et des remblais
- Fourier : systèmes d’équations linéaires (→ programmation linéaire)
- Borel, Von Neumann : théorie des jeux
- Sainte-Lague, König : théorie des graphes
- 1956 : Recherche opérationnelle civile (les militaires s’en étaient appropriés en premier), sorte d’“aide à la décision”

Domaines

1. Combinatoire
2. Aléatoire
3. Concurrentiel

Exemple 1. Problème d’affectation : on a n personnes et n postes. Chaque personne inscrit sa préférence pour un certain poste dans une grille. Comment trouver l’affectation qui maximise la somme des préférences ?

Il y a $n!$ solutions ; pour $n = 4$ cela en fait 24 ; pour $n = 20$, on atteint $8 \cdot 10^{10}$ solutions.

Un problème *facile* est un problème dont le nombre d'opérations est borné par un polynôme.

Un problème *difficile* est un problème pour lequel il n'y a pas d'algorithme polynomial connu ; on conjecture qu'il n'en existe pas.

Exemple 2. Rotation d'avions, d'équipages. . .

Exemple 3. Problème du *voyageur de commerce* : traverser les villes d'un pays et retourner à son point de départ en minimisant le coût (trouver un *circuit hamiltonien de valeur minimale*). Problème NP-difficile ; pour n villes il y a $(n - 1)!$ chemins différents.

Types de problèmes étudiés

Programmation linéaire

Il s'agit d'optimiser une fonction linéaire :

$$\left\{ \begin{array}{l} \min cx \\ Ax = b \\ x \geq 0 \end{array} \right\} \iff \left\{ \begin{array}{l} \min \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j = b_i \quad i \in \llbracket 1, m \rrbracket \\ x_i \geq 0 \quad i \in \llbracket 1, m \rrbracket \end{array} \right.$$

Programmation linéaire en nombre entiers

On remplace la contrainte $x \geq 0$ par $\forall i \in \llbracket 1, m \rrbracket, x_i \in \mathbf{N}$, voire $x_i \in \{0, 1\}$ (dans le cas de prises de décisions).

Aléatoire/Stochastique

Problèmes de files d'attente, de fiabilité et de gestion des stocks. Par exemple, on souhaite étudier le temps moyen d'attente à une file d'attente, et savoir combien de guichets supplémentaires on doit ouvrir pour ramener ce temps moyen à une certaine valeur.

Chapitre 1

Programmation dynamique

1.1 Généralités

Principe d'optimalité de Bellman : Toute sous-politique d'une politique optimale est optimale.

Par exemple, lorsqu'on a trouvé un chemin optimal d'un sommet A vers un sommet B , et que les sommets C et D sont sur ce même chemin, le chemin optimal de C vers D est le sous-chemin qui emprunte une partie de celui de A vers B .

1.2 Exemple : Tracé optimal d'une autoroute

On se donne le graphe en figure 1.1. On veut déterminer le tracé de coût minimal ; tracé qui s'exprime par

$$\text{coût d'un tracé} = \sum \text{coûts des tronçons}$$

Algorithme possible :

1. Chemin optimal depuis A jusqu'à chaque ville de la phase 2 :

x_2	chemin optimal	coût
E	ACE, ADE	10
F	ACF	9
G	ADG	9
H	ACH	9

2. Chemin optimal depuis A jusqu'à chaque ville de la phase 3 :

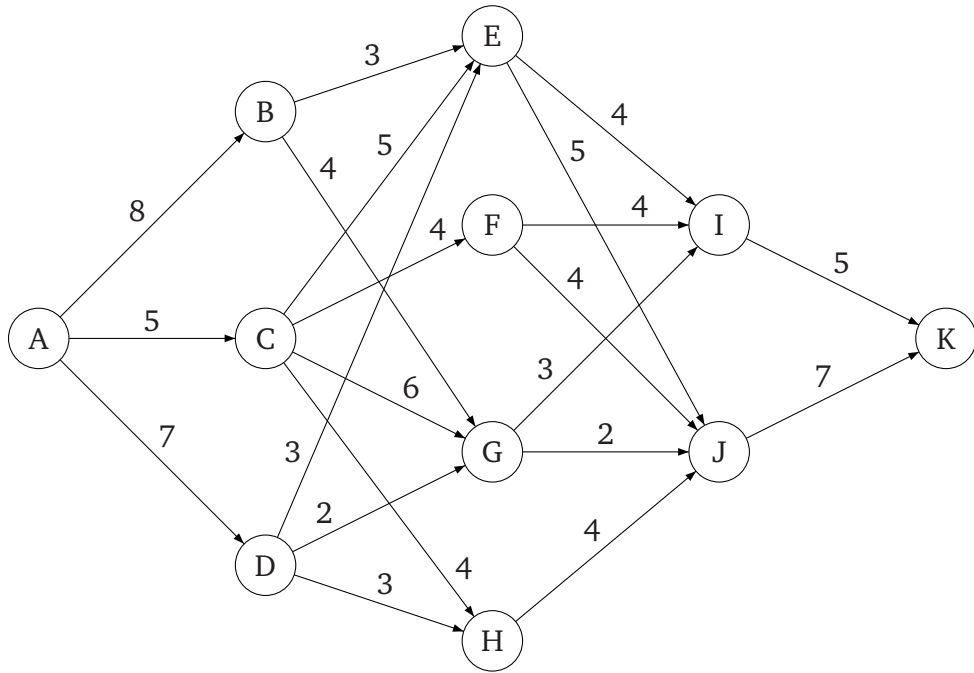


FIGURE 1.1 – Tracé d’une autoroute.

x_3	chemin optimal	coût
I	$A \cdots GI$	12
J	$A \cdots GJ$	11

3. Chemin optimal depuis A jusqu’à chaque ville de la phase 4 :

x_4	chemin optimal	coût
K	$A \cdots IK$	17

4. En considérant chaque possibilité de la fin vers le début, on en déduit que le chemin optimal est $ADGIK$ de coût 17.

1.3 Formalisation

Soit F une fonction de $N + 1$ variables à optimiser :

$$F(x_0, x_1, \dots, x_N) = v_1(x_0, x_1) + v_2(x_1, x_2) + \cdots + v_N(x_{N-1}, x_N)$$

Pour tout $n \in \llbracket 0, N - 1 \rrbracket$, x_n dépend de x_0 et de x_{n+1} .

Dans notre exemple précédent, nous avons :

$$F(x_0, x_1, \dots, x_n) = \sum_{i=1}^4 v_i(x_{i-1}, x_i)$$

où $v_i(x_{i-1}, x_i)$ est le coût du tronçon reliant la ville x_{i-1} de la phase $i - 1$ à la ville x_i de la phase i .

1.3.1 Formules d'optimisation séquentielle

On cherche $f_2(x_0, x_2)$, la valeur optimale de $v_1(x_0, x_1) + v_2(x_1, x_2)$, c'est-à-dire

$$f_2(x_0, x_2) = \min_{x_1 \in X_1(x_0, x_2)} (v_1(x_0, x_1) + v_2(x_1, x_2))$$

Ensuite, on cherche la valeur optimale de $f_3(x_0, x_3)$, i.e.

$$f_3(x_0, x_3) = \min_{x_2} (f_2(x_0, x_2) + v_3(x_2, x_3))$$

On continue ensuite jusqu'à trouver $f_N(x_0, x_N)$. Ainsi, l'optimum recherché est obtenu avec la formule

$$\min F(x_0, x_1, \dots, x_N) = \min_{x_0, x_N} f_N(x_0, x_N) \quad (1.1)$$

Dans notre exemple :

$$\begin{aligned} f_2(x_0, E) &= \min_{x_1 \in \{B, C, D\}} (v_1(A, x_1) + v_2(x_1, E)) = 10 \\ f_3(A, I) &= \min_{x_2 \in \{E, F, G\}} (f_2(A, x_2) + v_3(x_2, I)) = 12 \\ f_4(A, K) &= \min_{x_3 \in \{I, J\}} (f_3(x_0, x_3) + v_4(x_3, K)) = 17 \end{aligned}$$

1.4 Deuxième exemple d'application

On se propose de faire une campagne de publicité dans quatre régions. On possède un budget de B unités. On investit un certain nombre d'unités dans chaque région ; le rendement étant donné dans le tableau suivant :

	I	II	III	IV
0	0	0	0	0
1	0,28	0,25	0,15	0,20
2	0,45	0,41	0,25	0,35
3	0,65	0,55	0,40	0,42
4	0,78	0,65	0,50	0,48
5	0,90	0,75	0,62	0,53

Déterminer la stratégie optimale d'investissement.

Exemple de stratégie d'investissement :

	I	II	III	IV
1	0	2	2	
0,28	0	0,25	0,35	

ce qui donne un rendement de 0,88.

Si on tente de rechercher $\max F(x_1, x_2, x_3, x_4)$ où x_i est la somme investie dans la région i , ça ne marchera pas. Il faut rechercher $\max F(x_1, x_2, x_3, x_4)$ où x_i est le *sous-total* des sommes investies dans les régions $1, 2, \dots, i$. On pose alors $v_i(x_{i-1}, x_i)$ le rendement obtenu en investissant $x_i - x_{i-1}$ dans la région i .

On a

$$f_2(x_2) = \max_{x_1 \in \llbracket 0, x_2 \rrbracket} (v_1(x_0, x_1) + v_2(x_1, x_2))$$

le rendement maximal que l'on peut obtenir en investissant x_2 unités au total dans les deux premières régions. Il faut faire les calculs pour tous les cas possibles pour x_2 :

x_2	$f_2(x_2)$	x_1
0	0	0
1	0,28	1
2	0,53	1
3	0,70	2
4	0,90	3
5	1,06	3

Au rang 3, on a :

$$\begin{aligned} f_3(x_3) &= \text{rendement max en investissant } x_3 \text{ dans I, II et III} \\ &= \max_{x_1, x_2} (v_1(x_0, x_1) + v_2(x_1, x_2) + v_3(x_2, x_3)) \\ &= \max_{x_2 \in \llbracket 0, x_3 \rrbracket} (f_2(x_2) + v_3(x_2, x_3)) \end{aligned}$$

d'où

	(I, II)	III	x_3	$f_3(x_3)$	x_2
0	0	0	0	0	
1	0,28	0,15	1	0,28	1
2	0,53	0,25	2	0,53	2
3	0,70	0,40	3	0,70	3
4	0,90	0,50	4	0,90	4
5	1,06	0,62	5	1,06	5

L'optimum est atteint lorsque $x_3 = 0$; cette région n'est pas suffisamment intéressante pour investir dedans. Passons ensuite au rang 4 :

x_4	$f_4(x_4)$	x_3
5	1,10	4

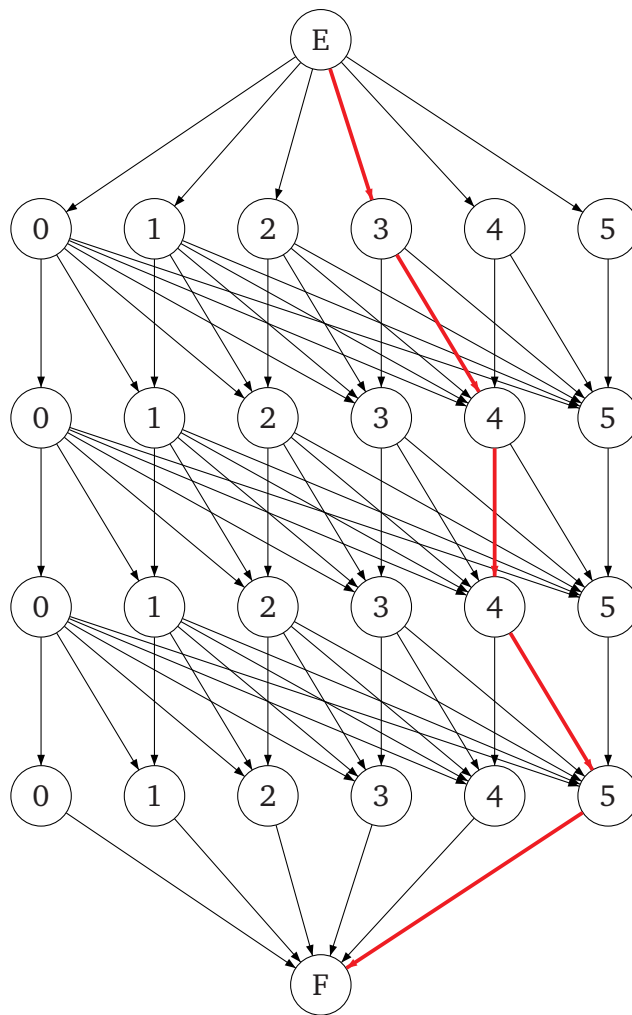
L'optimum est atteint lorsque $x_4 = 4$.

La solution est donc la suivante :

	I	II	III	IV	Total
Investissement cumulé	3	4	4	5	5
Investissement	3	1	0	1	5
Rendement	0,65	0,25	0	0,20	1,10

1.4.1 Représentation du problème en tant que graphe

Le problème peut également être représenté selon un graphe (fig. 1.2).



Chaque arc a pour valeur le rendement de l'investissement envisagé.

FIGURE 1.2 – Graphe des investissements

Ici, il faut trouver le chemin de valeur *maximale* de E à F .

Chapitre 2

Chemins optimaux dans les graphes

2.1 Position du problème

Soit $G = (X, U)$ un graphe. Pour tout $u \in U$, on note $v(u)$ la valeur de l'arc u . Cette valeur peut représenter une distance, un temps, ou de façon générale un coût.

Soient $(A, B) \in X^2$. La valeur du chemin μ est simplement :

$$\sum_{u \in \mu} v(u)$$

Problème : déterminer un chemin de valeur minimal de A à B .

Exemple 4. Réseau de télécommunications : chaque couple (A, B) de stations est reliée par un nombre donné de stations intermédiaires. Chaque arc a alors pour valeur $p_{i,j}$, qui est la probabilité de fonctionnement correct de la liaison entre i et j .

Trouver le chemin μ de A à B consiste à trouver :

$$\max \prod_{(i,j) \in \mu} p_{i,j}$$

ou encore

$$\min \sum_{(i,j) \in \mu} -\log(p_{i,j})$$

2.2 Algorithme de Dijkstra

Soit $G = (X, U)$ quelconque, tel que $\forall u \in U, v(u) \geq 0$. s est une racine. On attribue pour chaque $x \in X$ la valeur $\pi(x)$. La valeur $\pi(x)$ peut être :

- *Définitive* : valeur du chemin optimal de s à x .
- *Provisoires* : borne supérieure de la valeur du chemin optimal de s à x .

On note $c(s, x)$ la valeur de l'arc (s, x) s'il existe, $+\infty$ sinon.

Algorithme 1 Algorithme de Dijkstra

```

{Initialisation}
 $\pi(s) = 0$ 
pour chaque  $x \neq s$  faire
     $\pi(x) = c(s, x)$ 
     $A(x) = s$ 
fin pour
 $D = \{s\}$ 
 $P = X - \{s\}$ 
 $A(s) = \varepsilon$ 

tant que  $P \neq \emptyset$  faire
    {Attribution d'une valeur définitive à un sommet de  $P$ }
    Soit  $x_0 \in P$  t.q.  $\pi(x_0) = \min_{x \in P} \pi(x)$ 
     $D = D + \{x_0\}$ 
     $P = P - \{x_0\}$ 

    {Révision éventuelle des valeurs attribuées au sommet de  $P$ }
    pour chaque  $x \in P$  faire
        si  $\pi(x) > \pi(x_0) + c(x_0, x)$  alors
             $\pi(x) = \pi(x_0) + c(x_0, x)$ 
             $A(x) = x_0$ 
        fin si
    fin pour
fin tant que

```

La complexité de l'algorithme est $O(n^2)$.

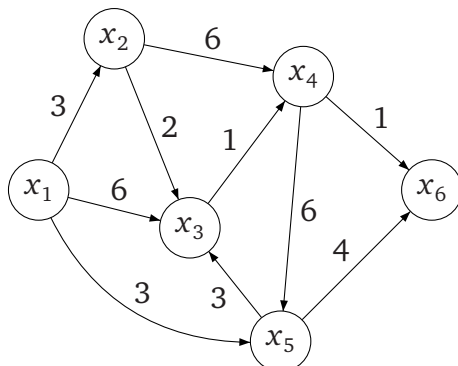


FIGURE 2.1 – Exemple d’application de l’algorithme de Dijkstra

2.2.1 Exemple d’application

	x_1	x_2	x_3	x_4	x_5	x_6
Init.	0	3	6	∞	3	∞
	0	3	6	9	3	∞
	0	3	5	9	3	8
	0	3	5	6	3	8
	0	3	5	6	3	7
	0	3	5	6	3	7

2.2.2 Preuve de l’algorithme

Démonstration. Soit k le numéro de l’étape de l’algorithme. À chaque étape k , pour tout $x \in D$, $\pi(x)$ a pour valeur celle du chemin optimal de s à x .

On sélectionne x_0 . Montrer que $\pi(x_0)$ égale la valeur du chemin optimal de s à x .

Soit $\pi(x_0)$ la valeur d’un chemin de s à x_0 . Soit μ un chemin quelconque de s à x_0 . Soit h le premier sommet de P rencontré, tel que x est prédécesseur de h , et h fait partie du chemin entre μ .

On pose μ_1 le chemin de s à h , μ_2 le chemin de h à x_0 .

La phase de révision des valeurs après le choix de x donne :

$$\begin{aligned}
 \pi(h) &\leq \pi(x) + c(x, h) \\
 \pi(x) + c(x, h) &\leq c(\mu_1) & \pi(x) &= \text{val du chemin opt. de } s \text{ à } x \\
 \pi(x_0) &\leq \pi(h) & & \text{par définition de } x_0
 \end{aligned}$$

Et ainsi, par transitivité :

$$\pi(x_0) \leq c(\mu_1) \leq c(\mu) \quad \square$$

2.3 Algorithme de Bellman–Ford

Soit $G = (X, U)$ un graphe orienté quelconque, avec n sommets, sans circuit absorbant, la valeur des arcs étant quelconque ; soit s une racine.

Algorithme 2 Algorithme de Bellman–Ford

{Initialisation}

$$\pi^1(s) = 0$$

$$\pi^1(x) = c(s, x) \text{ pour } x \neq s$$

{Étape générale : calcul des $\pi^{m+1}(x)$ en fonction des $\pi^{m-1}(x)$ }.

pour $m \in \llbracket 1, n - 2 \rrbracket$ **faire**

$$\pi^{m+1}(x) = \min\{\pi^m(x), \min_{y \neq x} \{\pi^m(y) + c(y, x)\}\}$$

fin pour

$$\pi^{n-1}(x) = \text{valeur du chemin optimal de } s \text{ à } x$$

{Hypothèse de récurrence}

$$\pi^m(x) = \text{valeur du chemin opt. de } s \text{ à } x \text{ de longueur } \leq m$$

Cet algorithme a pour complexité $O(n^3)$.

2.3.1 Exemple d'application

	x_1	x_2	x_3	x_4	x_5	x_6
$\pi^1(x)$	0	3	6	$+\infty$	3	$+\infty$
$\pi^2(x)$	0	3	0	7	3	8
$\pi^3(x)$	0	3	0	1	3	8
$\pi^4(x)$	0	3	0	1	3	2
$\pi^5(x)$	0	3	0	1	3	2

On peut s'arrêter dès qu'il y a deux lignes identiques.

Trouver le chemin optimal de x_i à x_j dans un graphe quelconque sans circuit absorbant : appliquer l'algorithme 2 en prenant pour origine x_1 puis x_2 , etc.

Cet algorithme a pour complexité $O(x^4)$.

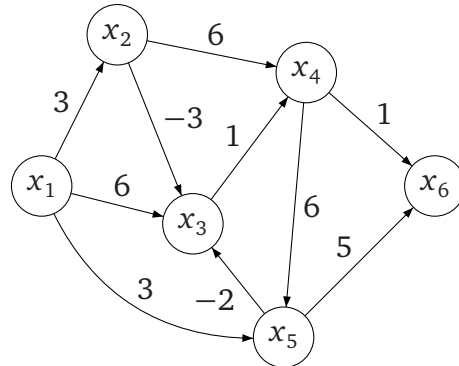


FIGURE 2.2 – Exemple d’application de l’algorithme de Bellman-Ford

2.4 Algorithme de Floyd–Warshall

Algorithme 3 Algorithme de Floyd–Warshall

{Initialisation}

$\pi_{i,j}^{(0)}$ valeur de (i, j) s’il existe, $+\infty$ sinon

pour $m = 1, 2, \dots, n$ **faire**

pour chaque (i, j) **faire**

$\pi_{ij}^{(m)} = \min \left(\pi_{ij}^{(m-1)}; \pi_{i,m}^{(m-1)} + \pi_{m,j}^{(m-1)} \right)$

fin pour

fin pour

On a alors $\pi_{i,j}^{(n)}$ la valeur du chemin optimal de i à j , et $\pi_{i,j}^{(m)}$ la valeur du chemin optimal de i à j dont les sommets intermédiaires appartiennent à $\llbracket 1, m \rrbracket$.

Cet algorithme a pour complexité $O(n^3)$.

Chapitre 3

Flots dans les graphes

3.1 Généralités

Il ne s'agit plus de trouver un chemin, mais de savoir quelle quantité de matière on peut faire circuler entre deux sommets.

Soit $G = (X, U)$. Soit $(s, t) \in X^2$ un couple particulier de sommets ; on appelle s la *source*, et t le *puits*. Pour tout $(i, j) \in U$, on note c_{ij} la *capacité* de l'arc, tel que $c_{ij} \geq 0$.

Flot Pour chaque arc $(i, j) \in U$, il faut associer une valeur x_{ij} appelée *flux* sur l'arc (i, j) , tel que $0 \leq x_{ij} \leq c_{ij}$.

La loi de conservation des flux stipule que

$$\forall i \in X \setminus \{s, t\}, \sum_{j \in \Gamma^-(i)} x_{ji} = \sum_{j \in \Gamma^+(i)} x_{ij}$$

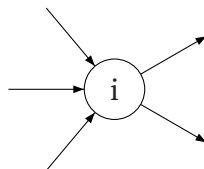


FIGURE 3.1 – Loi de conservation sur un nœud

Propriété 1. Soit $A \subset X$ un sous-graphe tel que $s, t \notin A$. Alors

$$\sum_{(i,j) \in \omega^-(A)} x_{ij} = \sum_{(i,j) \in \omega^+(A)} x_{ji}$$

La loi de conservation ne s'applique pas qu'à des nœuds, mais aussi à des sous-graphes entiers.

On pose

$$v = \sum_{j \in \Gamma^-(t)} x_{jt} - \sum_{j \in \Gamma^+(t)} x_{tj}$$

La quantité v correspond au flux total qui sort du réseau.

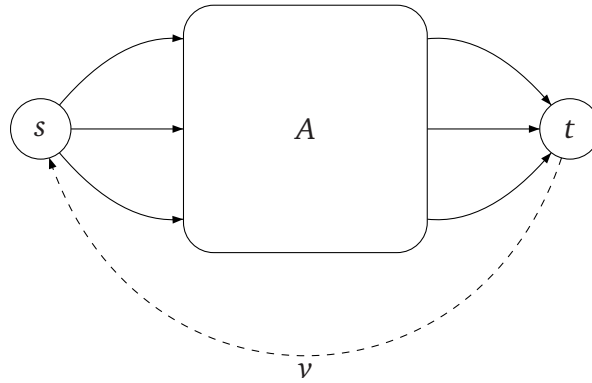


FIGURE 3.2 – Graphe avec arc de retour

En écrivant la loi de conservation entre s et t :

$$\sum_{j \in \Gamma^+(s)} x_{sj} + \sum_{j \in \Gamma^-(t)} x_{tj} = \sum_{j \in \Gamma^-(s)} x_{js} + \sum_{j \in \Gamma^+(t)} x_{jt}$$

d'où

$$v = \sum_{j \in \Gamma^+(s)} x_{sj} - \sum_{j \in \Gamma^-(s)} x_{js}$$

On ajoute alors un *arc fictif de retour* de flux v afin que la loi de conservation soit valable en tous les sommets. La valeur v est appelée *valeur du flot* (de s à t).

Exemple 5. Sur la figure 3.3, la valeur du flot est 6. Mais peut-on faire mieux ?

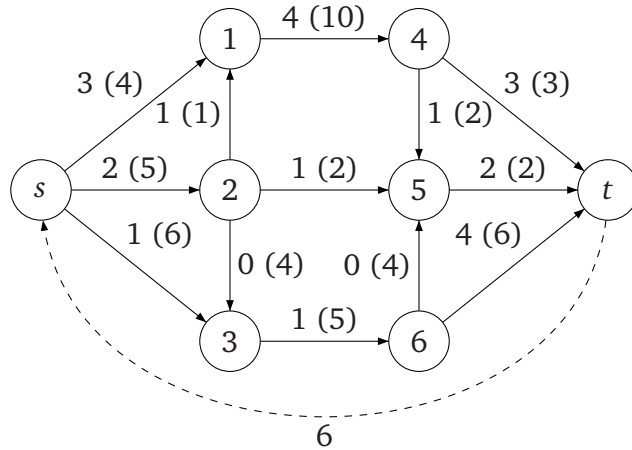


FIGURE 3.3 – Exemple de calcul de ν

3.2 Problème du flot maximal

On cherche à maximiser ν sous les contraintes :

$$\left\{ \begin{array}{ll} \max \nu = x_{ts} & \text{Flot maximal} \\ \sum_{j \in \Gamma^+(i)} x_{ij} = \sum_{j \in \Gamma^-(i)} x_{ji} & \text{Loi de conservation} \\ 0 \leq x_{ij} \leq c_{ij} & \end{array} \right.$$

3.2.1 Définitions

Définition 1. Un *flot admissible* est un flux qui vérifie les contraintes (mais non nécessairement optimal).

Définition 2. Un *flot complet* est un flot admissible tel que tous les chemins de s à t comportent au moins un arc saturé.

Définition 3. Un arc (i, j) est *saturé* ssi $x_{ij} = c_{ij}$.

Définition 4. Soient $(i, j) \in X^2$. Une chaîne entre i et j est dite *augmentante* si tous les arcs orientés de i vers j sont non saturés et si tous les arcs orientés de j vers i ont un flux strictement positif (i.e. non nul).

Autrement dit : Soit μ une chaîne entre s et t ; on note μ^+ les arcs orientés de s vers t , μ^- ceux orientés de t vers s . Alors :

$$\begin{array}{ll} x_{ij} < c_{ij} & \forall (i, j) \in \mu^+ \\ x_{ij} > 0 & \forall (i, j) \in \mu^- \end{array}$$

On va pouvoir augmenter le flot en modifiant uniquement les arcs de μ .

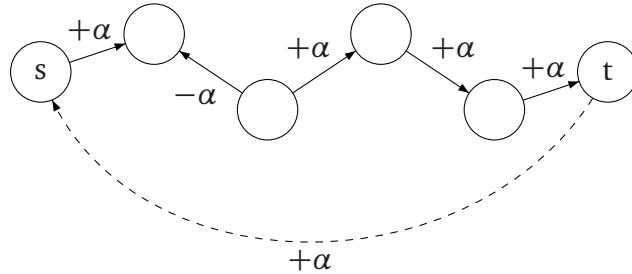


FIGURE 3.4 – Augmentation de la valeur du flot de α

Exemple 6. L'augmentation du flot consiste à :

- Ajouter α sur tous les arcs de μ^+
- Retrancher α sur tous les arcs de μ^- .

Ceci permet d'augmenter la valeur du flot de α .

La valeur de α maximale est

$$\alpha = \min \left[\min_{(i,j) \in \mu^+} (c_{ij} - x_{ij}); \min_{(i,j) \in \mu^-} x_{ij} \right]$$

Définition 5. Une *coupe* notée (S, T) est une partition de X en deux, tel que $s \in S$ et $t \in T$.

Définition 6. On appelle *capacité* de (S, T) la grandeur

$$c(S, T) = \sum_{\substack{i \in S, j \in T \\ (i,j) \in U}} c_{ij}$$

Propriété 2. Soit x un flot admissible, et (S, T) une coupe du graphe. Alors, la valeur du flot admissible est inférieure ou égale à $c(S, T)$. On a égalité lorsque le flot est maximal.

Démonstration. D'après la loi de conservation appliquée en T :

$$\sum_{(i,j) \in \omega^-(T)} x_{ij} = \sum_{(i,j) \in \omega^+(T)} x_{ij} \geq v$$

Or, par définition de $c(S, T)$, on a

$$c(S, T) = \sum_{(i,j) \in \omega^-(T)} c_{ij}$$

D'après ces deux égalités, on a

$$v \leq \sum_{(i,j) \in \omega^-(T)} c_{ij} \quad \square$$

3.2.2 Condition nécessaire et suffisante d'optimalité

Théorème 3 (Condition nécessaire et suffisante d'optimalité). *Un flot de s à t est maximal ssi il n'admet pas de chaîne augmentante de s à t .*

Démonstration. Condition nécessaire S'il existait une chaîne augmentante, on peut augmenter de α , donc le flot ne serait pas optimal.

Condition suffisante Le flot x n'admet pas de chaîne augmentante de s à t . Définissons une coupe telle que S est l'ensemble des sommets i du graphe tel qu'il existe une chaîne augmentante de s à i , et T l'ensemble complémentaire. Ainsi, $s \in S$ et $t \in T$.

À la "frontière" entre S et T , on constate alors que

$$\forall (i, j) \in U \text{ t.q. } i \in S, j \in T, x_{ij} = c_{ij}$$

et

$$\forall (i, j) \in U \text{ t.q. } i \in T, j \in S, x_{ij} = 0$$

L'écriture de la loi de conservation en T donne :

$$\sum_{(i,j) \in \omega^-(T)} x_{ij} = \sum_{(i,j) \in \omega^+(T)} x_{ij}$$

d'où

$$\sum_{(i,j) \in \omega^-(T)} c_{ij} = v$$

Le flot x est donc bien maximal, et (S, T) est une coupe de capacité minimale. \square

Théorème 4 (de Ford–Fulkerson). *La valeur maximale d'un flot est égale à la capacité minimale d'une coupe.*

Ce théorème illustre bien la notion de *goulet d'étranglement*.

3.3 Algorithme de Ford–Fulkerson

3.3.1 L'algorithme

Soit $G = (X, U)$, $(s, t) \in X^2$.

Algorithme 4 Algorithme de Ford–Fulkerson

{Initialisation}

Marquer \oplus le sommet s

{Étape générale}

si i marqué, j non marqué et (i, j) non saturé **alors**

 Marquer j avec \oplus_i

fin si

si i marqué, j non marqué et (j, i) t.q. $x_{ij} > 0$ **alors**

 Marquer j avec \ominus_i

fin si

si on ne peut plus marquer de sommets **alors**

si t est marqué **alors**

 {il existe une chaîne augmentante de s à t }

 augmenter le flot de α

 recommencer

sinon

 le flot considéré est maximal

fin si

fin si

3.3.2 Exemple d'application

On part d'un réseau et d'un flot initial. Une première application de l'algorithme est donnée dans la figure 3.5.

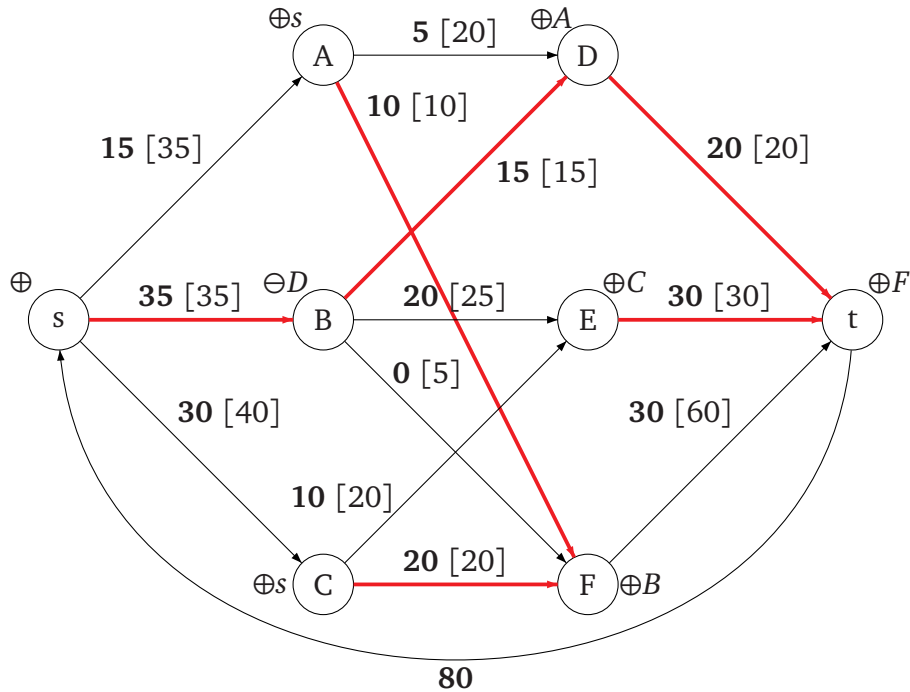


FIGURE 3.5 – Première application de l'algorithme

On constate que t est marqué. Cela signifie qu'on a une chaîne augmentante ; celle-ci est explicitée dans la figure 3.6.

L'arc (B, F) est celui qui va nous limiter. Ainsi, on a le flot donné en figure 3.7.

En reprenant l'algorithme, on ne peut plus marquer F , et donc on ne peut plus trouver t . Le flot maximal a donc pour valeur $v = 85$.

La coupe de capacité minimale est

$$S = \{\text{sommets marqués}\} \text{ et } T = \{\text{sommets non marqués}\}$$

donc ici, $S = \{s, A, B, C, D, E\}$ et $T = \{F, t\}$.

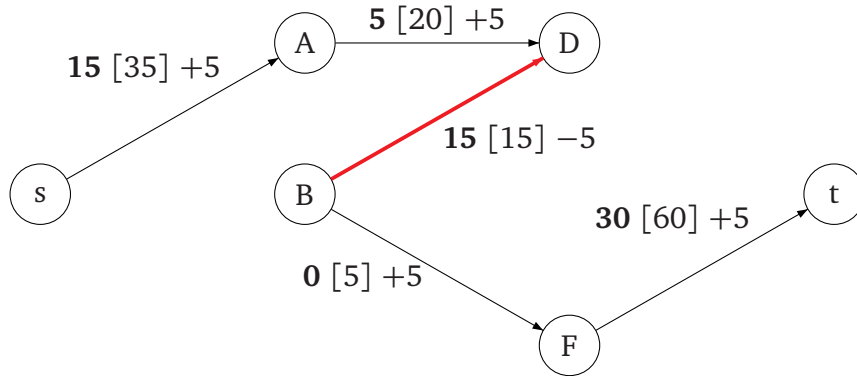


FIGURE 3.6 – Chaîne augmentante trouvée par l’algorithme

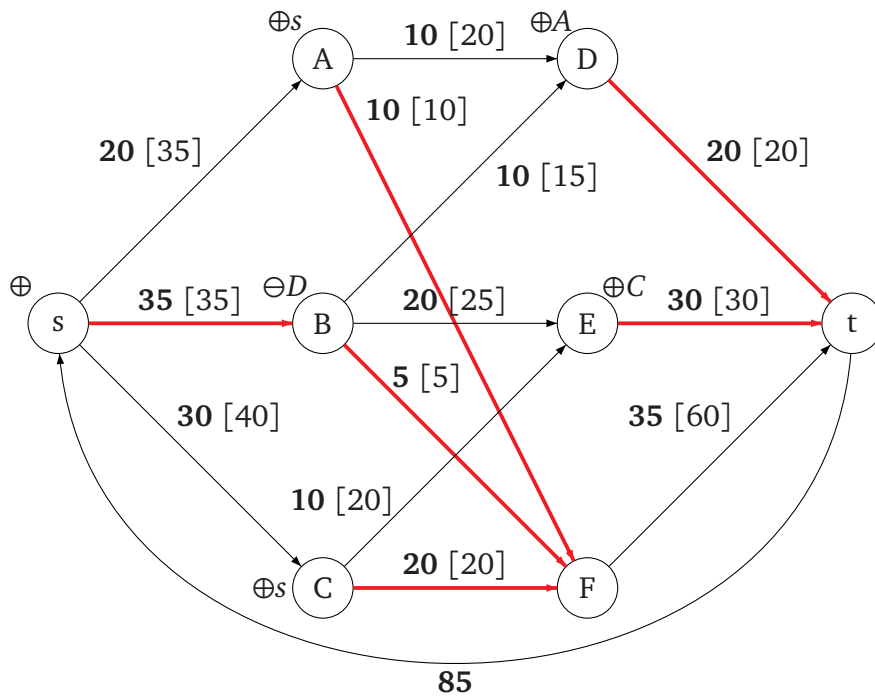


FIGURE 3.7 – Flot obtenu à l’issue de l’algorithme

Chapitre 4

Ordonnancement

4.1 Généralités

On se donne un projet avec de multiples tâches, et on souhaite organiser le mieux possible l'exécution de ces tâches.

Cette application de la recherche opérationnelle a vu le jour dans les années 1960, avec la méthode PERT (*Program Evaluation and Review Technique*) et la méthode des potentiels.

Les critères d'optimisation sont les suivantes :

- la *date d'achèvement du projet* : on va essayer de terminer le plus tôt possible ;
- l'*utilisation des ressources* : on va essayer de "lisser" l'utilisation des ressources (personnels, matériels, financiers...) afin de ne pas provoquer trop d'"à-coups".

On a alors les types de contraintes suivants :

1. Contraintes *potentielles* :
 - Contrainte d'*antériorité* : la tâche i ne peut commencer que quand j est achevée) ;
 - *Localisation temporelle* : la tâche i ne peut commencer avant une date d ou doit être terminée avant la date d' .
2. Contraintes *cumulatives* (sur les moyens) : respect les moyens disponibles ;
3. Contraintes *disjonctives* : les tâches i et j ne peuvent être réalisées simultanément (pour des raisons de sécurité, par exemple).

Le but est donc de déterminer un *ordre* et un *calendrier* d'exécution des tâches, qui minimise la durée totale de réalisation du projet.

4.2 La méthode PERT

Un graphe PERT est un graphe où les sommets sont des *événements*, i.e. le démarrage ou la réalisation d'un ensemble de tâches. Les arcs sont des *tâches* auxquelles on associe une durée.

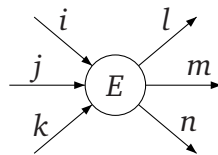


FIGURE 4.1 – Représentation d'un événement en PERT

Dans l'exemple de la figure 4.1, on dit que l'événement E est *réalisé* lorsque les tâches incidentes intérieurement à E (ici i , j et k) sont terminées. De la même façon, les tâches incidentes extérieurement (ici l , m et n) ne peuvent commencer que si E est réalisé.

On utilise des *tâches fictives* (de durée nulle) afin de représenter des contraintes.

4.2.1 Exemple de diagramme

Tâche	Durée	Dépendances
a	1	(aucune)
b	2	a
c	1	b
d	1	c
e	2	a
f	1	e
g	2	f
h	2	f
i	1	g
j	2	h, i
k	1	d, j
l	2	k

La traduction de ce tableau donne alors le graphe en figure 4.2.

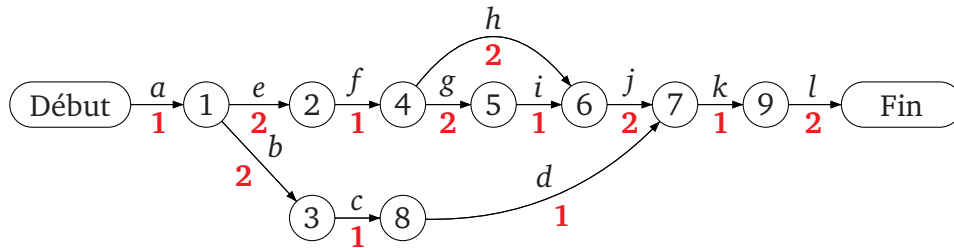


FIGURE 4.2 – Exemple de diagramme PERT

4.2.2 Exemple avec des tâches fictives

Tâche	Dépendances
<i>a</i>	(aucune)
<i>b</i>	(aucune)
<i>c</i>	<i>a</i>
<i>d</i>	<i>b</i>
<i>e</i>	<i>d</i>
<i>f</i>	<i>c, d</i>
<i>g</i>	<i>e, f</i>

On obtient alors le graphe en figure 4.3.

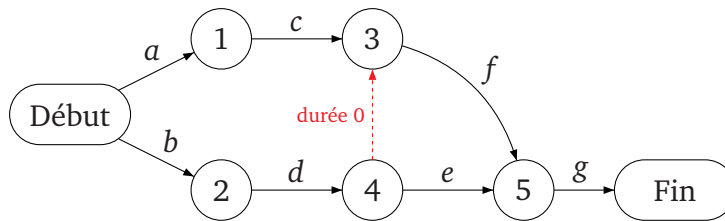


FIGURE 4.3 – Exemple de diagramme PERT avec des tâches virtuelles

4.2.3 Définitions

Définition 7. Soit i un événement. On appelle la *date au plus tôt* (notée t_i) la date la plus proche possible de réalisation de l'événement i . Autrement dit :

$$t_i = \max_{t_j \in \Gamma^-(i)} (t_j + d_{ji})$$

où d_{ji} est la durée de réalisation des tâches entre j et i .

On peut également dire que t_i est la valeur du chemin de valeur maximale du début à l'événement i .

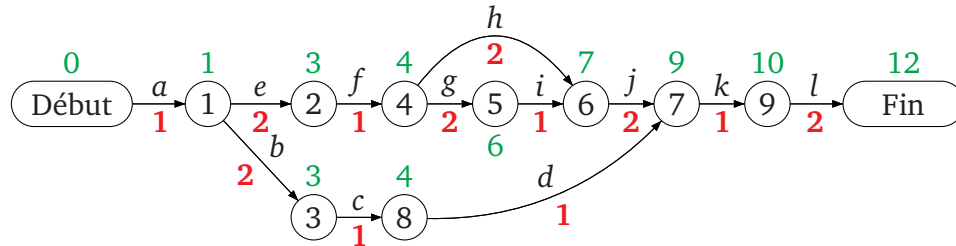


FIGURE 4.4 – Dates au plus tôt

Exemple 7. Les dates au plus tôt de chaque événement de la figure 4.4 sont notées en vert.

Définition 8. Un *chemin critique* est un chemin de valeur maximal du début à la fin.

Définition 9. On appelle *tâche critique* une tâche qui appartient au chemin critique.

Exemple 8. Sur la figure 4.2, les tâches critiques sont a, e, f, g, i, j, k, l .

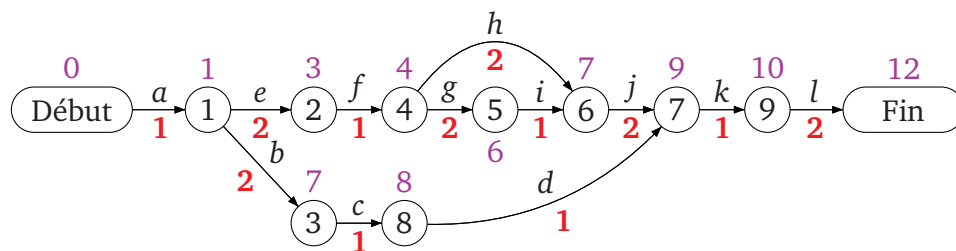


FIGURE 4.5 – Dates au plus tard

Définition 10. Soit i un événement. On appelle *date au plus tard*, notée t_i^* , la date la plus tardive de réalisation de l'événement i sans modification de la date au plus tôt de l'événement Fin.

Autrement dit :

$$t_i^* = t_{\text{Fin}} - \text{val. du chemin de val. max de } i \text{ à Fin}$$

ou encore

$$t_i^* = \min_{j \in \Gamma^+(i)} (t_j^* - d_{ij})$$

Exemple 9. Les dates au plus tard de la figure 4.5 sont représentées en violet.

4.2.4 Marges relatives aux tâches

Définition 11. La *marge libre* est le retard maximal qui peut être pris dans la réalisation de la tâche, sans modification de la date au plus tôt de l'événement terminal. Autrement dit :

$$m_{ij} = t_j - t_i - d_{ij}$$

Définition 12. La *marge totale* est le retard maximal sur la tâche qui ne modifie pas la date au plus tard de l'événement terminal. Autrement dit :

$$M_{ij} = t_j^* - t_i - d_{ij}$$

4.2.5 Exemple d'ordonnement d'atelier

On souhaite usiner des pièces de huit types différents. On dispose de deux machines, *A* et *B*. On donne les durées d'usinage a_i et b_i de chaque pièce par les deux machines.

Chaque pièce est usinée dans l'ordre et en deux étapes, d'abord par la machine *A*, puis par la machine *B*.

Pièces	a_i	b_i
P_1	5	7
P_2	6	4
P_3	9	8
P_4	7	5
P_5	8	10
P_6	6	6
P_7	4	3
P_8	5	6

Déterminons l'ordre optimal d'usinage des pièces.

On peut déjà déterminer les bornes inférieures et supérieures de durée totale :

$$b_{\text{inf}} = 53 \qquad b_{\text{sup}} = 60$$

On applique alors l'algorithme de Johnson.

Algorithme 5 Algorithme de Johnson

$L = 0, K = 0$

tant que il existe une pièce **faire**

choisir la pièce P_i t.q. $\min(a_i, b_i) \leq \min_{j \neq i}(a_j, b_j)$.

si $a_i < b_i$ **alors**

$K = K + 1$

$S(K) = i$

sinon

$L = L + 1$

$T(L) = i$

fin si

supprimer la pièce i

fin tant que

renvoyer ordre optimal $S(1), S(2), \dots, S(K), T(L), \dots, T(1)$

À l'issue de l'exécution de l'algorithme, on obtient :

$$S = (1, 8, 5) \qquad T = (7, 2, 4, 6, 3)$$

Il faut donc fabriquer les pièces dans l'ordre : 1, 8, 5, 3, 6, 4, 2, 7. Cet ordonnancement donne un temps total d'usinage de 54.

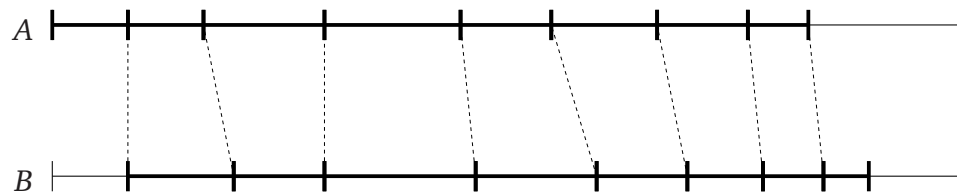


FIGURE 4.6 – Ordonnancement optimal

Chapitre 5

Méthodes d'énumération implicite

Aussi appelées :

- procédures arborescentes
- procédures par séparation et évaluation
- procédures *branch and bound*

5.1 Idée sur un exemple

Exemple 10. Soit $G = (X, E)$ un graphe. On veut déterminer dans G un sous-graphe complet (ou clique) de cardinal maximal.

Propriété 5. Soit un graphe avec n sommets et m arêtes. On note $\gamma(G)$ le cardinal maximal d'une clique. On peut montrer facilement que

$$\gamma(G) < \frac{1}{2} \left(3 + \sqrt{9 + 8(m - n)} \right)$$

Exemple 11. Dans le graphe de la figure 5.1, $m = 12$, $n = 7$ et donc $\gamma(G) \leq 5$.

Formalisation Soit S un ensemble fini, et $f : S \rightarrow \mathbf{R}$. On souhaite déterminer $s^* \in S$ tel que $f(s^*) = \min_{s \in S} f(s)$.

Définitions

Définition 13 (Évaluation). Étant donné $S' \subseteq S$, on dit qu'on sait évaluer S si on connaît un réel $g(S')$ tel que $\forall s \in S', g(S') \leq f(s)$.

Définition 14. Si $g(S') = \min_{s \in S'} f(s)$, il s'agit d'une *évaluation exacte*.

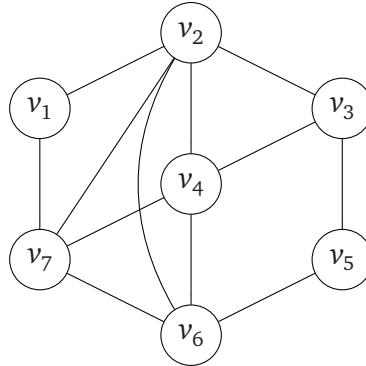


FIGURE 5.1 – Exemple de graphe

Définition 15. On dit que $g_1(S')$ est une *meilleure évaluation* que $g_2(S')$ si $g_1(S') > g_2(S')$.

Définition 16 (Abandon). On dit que $S' \subseteq S$ est *abandonné* dans les cas suivants :

- $S' = \emptyset$
- on connaît une solution $\hat{s} \in S$ au moins aussi bonne que toute solution de S' . Par exemple, lorsque $f(\hat{s}) \leq g(S')$.
- on a une évaluation exacte.

Propriété 6. Soit $F = \{S_1, S_2, \dots, S_p\}$ une famille de sous-ensemble de S dont l'union est égale à S . Si F est séparée en $S_i^1, S_i^2, \dots, S_i^k$, la famille F' telle que

$$F' = \{S_1, \dots, S_{i-1}, S_i^1, S_i^2, \dots, S_i^k, S_{i+1}, \dots, S_p\}$$

est encore une famille de sous-ensembles de S dont l'union vaut S si

$$\bigcup_{j=1}^k S_i^j = S_i$$

Propriété 7. Si tous les sous-ensembles de F ont une évaluation exacte, et si $S' \in F$ est d'évaluation minimale, alors la solution optimale de S' est la solution optimale du problème.

5.2 Schéma général

On note \hat{s}_i la meilleure solution connue, et v sa valeur. Le schéma général est donné dans l'algorithme 6.

Algorithme 6 Schéma général d'évaluation

\hat{s} indéterminé, $\nu = +\infty$, $F = \{S\}$.
tant que $F \neq \emptyset$ **faire**
 {Choix}
 choisir $S' \in F$ non encore évalué
 si un tel S' n'existe pas **alors**
 choisir un $S' \in F$ déjà évalué
 fin si
 si S' n'a pas été évalué **alors**
 {Évaluation}
 Associer à S' son évaluation $g(S')$
 si $g(S') \geq \nu$ **alors**
 $F = F - S'$
 sinon si l'évaluation est exacte **alors**
 \hat{s} est la solution opt. de S'
 $\nu = g(S')$
 $F = F - (S' \cup \{E \in F, g(E) \geq \nu\})$
 fin si
 sinon
 {Séparation}
 séparer S' en $(S'_i)_{i \in \llbracket 1, k \rrbracket}$
 $F = (F - S') \cup \bigcup_{i=1}^k S'_i$
 fin si
fin tant que

Il reste encore à spécifier les procédures de *choix*, d'*évaluation* et de *séparation*.

5.3 Application au problème du voyageur de commerce

Il s'agit de trouver le circuit hamiltonien de valeur minimale. Ce problème est difficile et est souvent utilisé comme test d'algorithme de résolution de problèmes combinatoires. En effet, dans un graphe à n sommets, il y a $(n-1)!$ circuits hamiltoniens.

On se donne un graphe complet orienté $G = (X, U)$. Pour tout arc $(i, j) \in U$, on associe la valeur $t_{ij} \geq 0$.

Dans cet exemple, prenons $n = 5$. Les coûts pour aller de la ville i à la ville j sont données par la matrice :

$$\begin{pmatrix} +\infty & 2 & 1 & 3 & 4 \\ 1 & +\infty & 6 & 5 & 7 \\ 4 & 3 & +\infty & 8 & 2 \\ 5 & 7 & 4 & +\infty & 1 \\ 2 & 3 & 5 & 2 & +\infty \end{pmatrix}$$

À chaque ligne, on soustrait les éléments de la ligne par la plus petite valeur de la ligne. On obtient alors :

$$\begin{pmatrix} +\infty & 1 & 0 & 2 & 3 \\ 0 & +\infty & 5 & 4 & 6 \\ 2 & 1 & +\infty & 6 & 0 \\ 4 & 6 & 3 & +\infty & 0 \\ 0 & 1 & 3 & 0 & +\infty \end{pmatrix}$$

On fait la même chose sur chaque colonne :

$$\begin{pmatrix} +\infty & 0 & 0 & 2 & 3 \\ 0 & +\infty & 5 & 4 & 6 \\ 2 & 0 & +\infty & 6 & 0 \\ 4 & 5 & 3 & +\infty & 0 \\ 0 & 0 & 3 & 0 & +\infty \end{pmatrix}$$

Propriété 8. La valeur de tout chemin hamiltonien est supérieure ou égale à la somme des éléments soustraits.

Démonstration. On pose x_{ij} telle que :

$$x_{ij} = \begin{cases} 1 & \text{ssi } (i, j) \in \text{chemin hamiltonien} \\ 0 & \text{sinon} \end{cases}$$

Alors, cette borne inférieure s'écrit :

$$\min \sum_i \sum_{j \neq i} t_{ij} x_{ij}$$

Or

$$\forall i, \sum_{j \neq i} x_{ij} = 1 \text{ et } \forall j, \sum_{i \neq j} x_{ij} = 1$$

On pose $t_{ij}^0 = t_{ij} - \alpha_i - \beta_j$, où α_i est la quantité soustraite à la ligne i , et β_j la quantité soustraite sur la colonne j . Formellement, on a :

$$\alpha_i = \min_j t_{ij} \qquad \beta_j = \min_i (t_{ij} - \alpha_i)$$

Par ailleurs, on constate que $t_{ij}^0 \geq 0$.

La valeur d'un circuit vaut alors :

$$\begin{aligned} & \sum_i \sum_{j \neq i} (t_{ij}^0 + \alpha_i + \beta_j) x_{ij} \\ &= \underbrace{\sum_i \sum_{j \neq i} t_{ij}^0 x_{ij}}_{\geq 0} + \sum_i \alpha_i \underbrace{\sum_{j \neq i} x_{ij}}_1 + \sum_j \beta_j \underbrace{\sum_{i \neq j} x_{ij}}_1 \end{aligned}$$

Ainsi, on peut écrire que la valeur de tout chemin hamiltonien est supérieur ou égal à

$$\sum \alpha_i + \sum \beta_j \quad \square$$

Dans notre exemple, cette borne inférieure vaut 8.

5.3.1 Calcul du regret

Le *regret* d'un arc (i, j) est la somme de la plus petite valeur de la ligne i et la plus petite valeur de la colonne j .

La séparation se fait sur l'arc de valeur 0 et de regret maximal. On calcule donc tous les regrets sur tous les arcs de valeur 0, et on constate que celui de regret maximal est l'arc (B, A) , qui vaut 4.

On supprime ensuite la ligne B et la colonne A de la matrice. On interdit également de prendre l'arc (A, B) . On continue les calculs avec la matrice

$$\begin{pmatrix} +\infty & 0 & 2 & 3 \\ 0 & +\infty & 6 & 0 \\ 5 & 3 & +\infty & 0 \\ 0 & 3 & 0 & +\infty \end{pmatrix}$$

Il se peut qu'on ait ensuite besoin de refaire les opérations de soustractions, auquel cas la borne inférieure V augmente d'autant que ce qui a été retranché. On obtient alors la séparation de la figure 5.2.

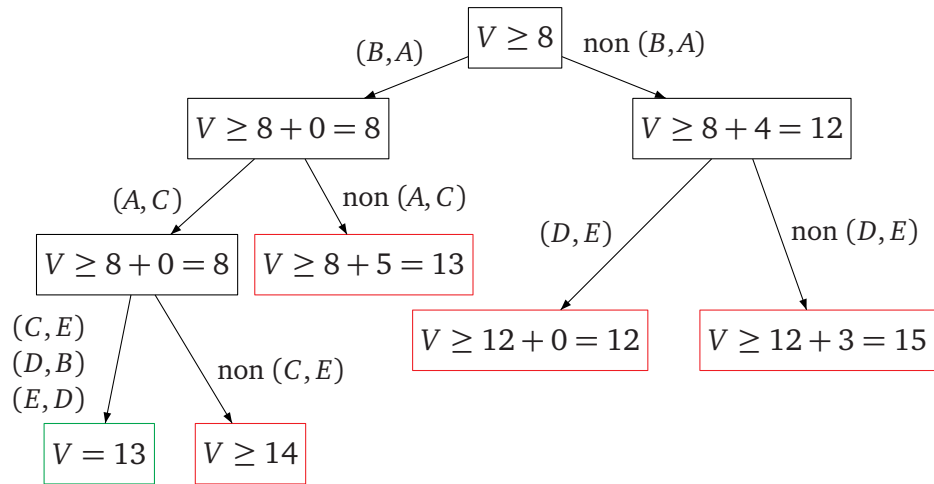


FIGURE 5.2 – Séparation

On choisit d'inclure l'arc (B,A) . Alors, en répétant le même procédé, on fait ensuite la séparation sur l'arc (A,C) , puis sur (C,E) , et ainsi de suite.

On trouve finalement un circuit optimal de valeur 12.

Chapitre 6

Programmation linéaire

On cherche à optimiser une fonction f linéaire sous des contraintes linéaires.

La forme générale d'un *programme linéaire* est la suivante :

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

sous les contraintes :

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

$$\sum_{j=1}^n a_{ij} x_j = b_i, i \in \bar{M}$$

avec M et \bar{M} un ensemble d'indices, $x_j \in \mathbf{R}_+$ pour $j \in M$ et $j \in \bar{M}$.

6.1 Problèmes de production

On a un *centre de production* disposant de *ressources* numérotées de 1 à m , et produisant n types de *produits*. On note :

- a_{ij} la quantité de la ressource j requise pour produire une unité du produit i ;
- x_i^{\min} (resp. x_i^{\max}) la quantité minimale (resp. maximale) de produit i à produire ;
- b_j la quantité de ressource j disponible ;
- m_i le bénéfice réalisé en produisant une unité de produit i ;
- x_i la quantité produite de i .

Le problème se formalise alors de la façon suivante :

$$\max \sum_{i=1}^n m_i x_i$$

s.c.

$$\begin{aligned} x_i &\leq x_i^{\max} & i \in \llbracket 1, n \rrbracket \\ x_i &\geq x_i^{\min} & i \in \llbracket 1, n \rrbracket \\ \sum_{i=1}^n a_{ij} x_i &\leq b_j & j \in \llbracket 1, m \rrbracket \\ x_i &\geq 0 & i \in \llbracket 1, n \rrbracket \end{aligned}$$

Ce n'est pas un programme linéaire. Mais on peut réécrire ce problème pour le ramener à un tel problème.

On applique alors l'algorithme du simplexe.

6.1.1 Optimisation fractionnaire

On considère le problème suivant :

$$\max r(x) = \frac{b_0 + \sum_{j=1}^n b_j x_j}{c_0 + \sum_{j=1}^n c_j x_j}$$

s.c.

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq d_i & i \in \llbracket 1, m \rrbracket \\ x_j &\geq 0 & j \in \llbracket 1, n \rrbracket \\ b_0, b_j, a_j &\geq 0 \\ c_0 &> 0 \end{aligned}$$

En posant

$$y_j = \frac{x_j}{c_0 + \sum_{j=1}^n c_j x_j} \quad t = \frac{1}{c_0 + \sum_{j=1}^n c_j x_j}$$

le problème se réécrit de la manière suivante :

$$\max b_0 t + \sum_{j=1}^n b_j y_j$$

s.c.

$$\begin{aligned} \sum_{j=1}^n a_{ij} y_j &\leq d_i t & i \in \llbracket 1, m \rrbracket \\ c_0 t + \sum_{j=1}^n c_j y_j &= 1 \\ y_j &\geq 0 & j \in \llbracket 1, n \rrbracket \\ t &\geq 0 \end{aligned}$$

6.1.2 Interprétation géométrique d'un programme linéaire

On considère le problème suivant :

$$\max x_1 + 3x_2$$

s.c.

$$x_1 + x_2 \leq 5$$

$$x_1 \leq 4$$

$$x_2 \leq 3$$

$$x_1, x_2 \geq 0$$

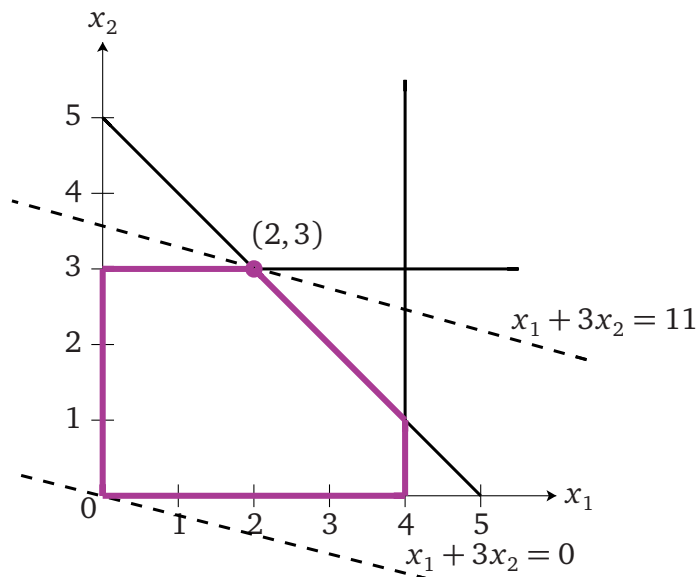


FIGURE 6.1 – Interprétation géométrique d'un programme linéaire

Le point optimal est $(2, 3)$; la valeur obtenue est 11.

6.1.3 Forme générale d'un programme linéaire

Le problème peut se présenter sous deux formes différentes :

Forme standard

$$\min cx$$

$$Ax = b$$

$$x \geq 0$$

Forme canonique

$$\min cx$$

$$Ax \geq b$$

$$x \geq 0$$

Afin de passer de la forme générale à la forme canonique, il faut remplacer la condition

$$\sum_{j=1}^n a_{ij}x_j = b_j$$

par les deux conditions

$$\left\{ \begin{array}{l} \sum_{j=1}^n a_{ij}x_j \geq b_i \\ \sum_{j=1}^n -a_{ij}x_j \geq -b_i \end{array} \right.$$

Afin de passer de la forme générale à la forme standard, il faut ajouter une *variable d'écart* $s_i \geq 0$. Ainsi, on transforme la condition

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

en

$$\sum_{j=1}^n a_{ij}x_j + s_i = b_i, s_i \geq 0$$

6.1.4 Base et solution de base

On suppose que $\text{rg}(A) = m$.

On appelle *matrice de base* la sous-matrice carrée ($m \times m$) de A régulière. Ainsi :

$$A = \begin{pmatrix} B & N \end{pmatrix} \text{ (par blocs)}$$

et

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}^{\mathcal{B}} \\ \mathbf{x}^{\mathcal{N}} \end{pmatrix} \text{ (par blocs)}$$

Les colonnes de B représentent les *variables de base*, et celles de N représentent les *variables hors base*.

D'où :

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{Bx}^{\mathcal{B}} + \mathbf{Nx}^{\mathcal{N}} &= \mathbf{b} \\ \mathbf{B}^{-1}\mathbf{Bx}^{\mathcal{B}} + \mathbf{B}^{-1}\mathbf{Nx}^{\mathcal{N}} &= \mathbf{B}^{-1}\mathbf{b} \end{aligned}$$

et donc

$$\mathbf{x}^{\mathcal{B}} + B^{-1}N\mathbf{x}^{\mathcal{N}} = B^{-1}\mathbf{b} \quad (6.1)$$

La solution de base relativement à B est la solution telle que $\mathbf{x}^{\mathcal{N}} = 0$ et $\mathbf{x}^{\mathcal{B}} = B^{-1}\mathbf{b}$.

Exemple 12. Considérons le problème suivant :

$$\min 3x_1 + x_2 + 4x_3 + 2x_4 - 2x_5$$

s.c.

$$\begin{aligned} x_1 + 2x_4 + 3x_5 &= 2 \\ 2x_2 + 4x_4 &= 6 \\ 2x_1 + x_2 + 2x_3 + 2x_5 &= 9 \\ x_1, x_2, \dots, x_5 &\geq 0 \end{aligned}$$

Ici, la matrice A vaut :

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 & 3 \\ 0 & 2 & 0 & 4 & 0 \\ 2 & 1 & 2 & 0 & 2 \end{pmatrix}$$

En utilisant les colonnes 1, 2, et 3 de A (car A est de rang 3 et ces colonnes sont linéairement indépendantes), on a :

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 1 & 2 \end{pmatrix} \quad B^{-1} = \frac{1}{4} \begin{pmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ -4 & -1 & 2 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 2 \\ 6 \\ 9 \end{pmatrix}$$

La solution de base réalisable est

$$x_1 = 2 \quad x_2 = 3 \quad x_3 = 1 \quad x_4 = 0 \quad x_5 = 0$$

de valeur 13.

L'ensemble $X = \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ est un polytope convexe.

Définition 17. On appelle *point extrême* tout point $x \in X$ qui ne peut être exprimé comme combinaison convexe d'autres points $y \in X$ ($y \neq x$).

Théorème 9. L'ensemble des points extrêmes de X correspond à l'ensemble des solutions de base réalisables.

Corollaire 10. X admet un nombre fini de points extrêmes. Le nombre de solutions de base est inférieur ou égal à C_n^m .

Corollaire 11. *Tout point d'un polytope convexe X peut s'exprimer comme une combinaison convexe des points extrêmes de X .*

Théorème 12 (Théorème central de la programmation linéaire). *L'optimum de z fonction linéaire sur le polytope convexe X est atteint en au moins un point extrême.*

S'il est atteint en plusieurs points extrêmes, alors il est atteint en tout point combinaison convexe de ces points extrêmes.

Démonstration. Soient y^1, y^2, \dots, y^K les points extrêmes de X .

Soit $z^* = \min_{k \in \llbracket 1, K \rrbracket} z(y^k)$. Montrons que z^* est le minimum de z sur X .

Soit $x \in X$. Alors

$$x = \sum_{k=1}^K \lambda_k y_k \quad \lambda_k \geq 0, \sum \lambda_k = 1$$

Par linéarité de z , on obtient

$$\begin{aligned} z(x) &= \sum_{k=1}^K \lambda_k z(y^k) \\ &\geq \sum_{k=1}^K \lambda_k z^* \\ &\geq z^* \end{aligned}$$

Donc z^* est la valeur de la solution optimale. □

6.2 Algorithme du simplexe

L'algorithme fait une exploration "dirigée" des points extrêmes.

6.2.1 Caractérisation des solutions de base réalisables optimales

On rappelle le problème décomposé en variables de base et hors base :

$$x^{\mathcal{B}} + B^{-1}N x^{\mathcal{N}} = B^{-1}b$$

Définition 18. Une solution de base est *dégénérée* si certaines variables de base sont nulles.

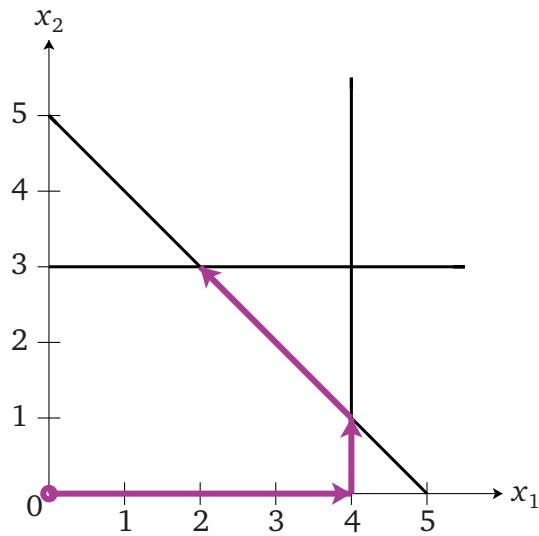


FIGURE 6.2 – Exploration dirigée d'un polytope

Théorème 13 (Condition nécessaire et suffisante d'optimalité). *La base \mathcal{B} non dégénérée est optimale ssi*

$$\forall j \in \mathcal{N}, \Delta_j = c_j - \sum_{i \in \mathcal{B}} c_i \bar{a}_{ij} \geq 0$$

Démonstration. Condition suffisante

Soit $i \in \mathcal{B}$. On a

$$x_i + \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j = \bar{b}_i$$

d'où

$$x_i = \bar{b}_i - \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j$$

Soit x une solution admissible. On a alors

$$\begin{aligned}
 z(x) &= \sum_{j=1}^n c_j x_j \\
 &= \sum_{i \in \mathcal{B}} c_i x_i + \sum_{j \in \mathcal{N}} c_j x_j \\
 &= \sum_{i \in \mathcal{B}} c_i \left(\bar{b}_i - \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j \right) + \sum_{j \in \mathcal{N}} c_j x_j \\
 &= \sum_{i \in \mathcal{B}} c_i \bar{b}_i + \sum_{j \in \mathcal{N}} x_j \left(c_j - \sum_{i \in \mathcal{B}} c_i \bar{a}_{ij} \right) \\
 &= \bar{z} + \sum_{j \in \mathcal{N}} \Delta_j x_j
 \end{aligned}$$

où

$$\bar{z} = \sum_{i \in \mathcal{B}} c_i \bar{b}_i \qquad \Delta_j = c_j - \sum_{i \in \mathcal{B}} c_i \bar{a}_{ij}$$

Si pour tout $j \in \mathcal{N}$, les Δ_j sont positifs ou nuls, alors toutes les solutions admissibles donnent à z une valeur supérieure ou égale à \bar{z} .

Condition nécessaire

Supposons $\Delta_s < 0$ pour un $s \in \mathcal{N}$. Soit $i \in \mathcal{B}$. Alors :

$$\begin{aligned}
 x_i &= \bar{b}_i - \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j \\
 &= \bar{b}_i - \sum_{\substack{j \in \mathcal{N} \\ j \neq s}} \bar{a}_{ij} x_j - \bar{a}_{is} x_s \\
 z &= \sum_{i \in \mathcal{B}} c_i \bar{b}_i + \sum_{\substack{j \in \mathcal{N} \\ j \neq s}} \Delta_j x_j + \Delta_s x_s
 \end{aligned}$$

Sans toucher aux autres variables hors base, on peut augmenter x_s d'une valeur $\hat{\theta}$ telle que :

$$\begin{aligned}
 \hat{\theta} &= \min_{\substack{i \in \mathcal{B} \\ \bar{a}_{is} > 0}} \frac{\bar{b}_i}{\bar{a}_{is}} \\
 &= \frac{\bar{b}_r}{\bar{a}_{rs}}
 \end{aligned}$$

La nouvelle solution de base est alors telle que :

$$\begin{aligned}x_s &= \hat{\theta} \\x_j &= 0 \quad j \in \mathcal{N}, j \neq s \\x_r &= 0 \\x_i &= \bar{b}_i - \bar{a}_{is} \quad i \in \mathcal{B}, i \neq r\end{aligned}$$

Ainsi :

$$\bar{z} = \sum_{i \in \mathcal{B}} c_i \bar{b}_i + \Delta_s \hat{\theta} \quad \square$$

Exemple 13.

$$\min Z(x) = 2x_1 - 3x_2 + 5x_3$$

s.c.

$$\begin{aligned}x_1 - 2x_2 + x_3 - x_4 &= 4 \\x_2 + 3x_3 + x_5 &= 6 \\2x_1 + x_3 + 2x_4 + x_6 &= 7 \\x_1, \dots, x_6 &\geq 0\end{aligned}$$

Ici, $\mathcal{B} = \{1, 3, 6\}$ et $\mathcal{N} = \{2, 4, 5\}$. Autrement dit :

$$\begin{aligned}B &= \begin{pmatrix} 1 & 1 & 0 \\ 0 & 3 & 0 \\ 2 & 1 & 1 \end{pmatrix} & B^{-1} &= \frac{1}{3} \begin{pmatrix} 3 & -1 & 0 \\ 0 & 3 & 0 \\ -6 & 1 & 3 \end{pmatrix} \\B^{-1}N &= \frac{1}{3} \begin{pmatrix} -7 & -3 & -1 \\ 3 & 0 & 1 \\ 13 & 12 & 1 \end{pmatrix} & B^{-1}\mathbf{b} &= \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}\end{aligned}$$

La solution de base est alors :

$$\begin{aligned}x_1 &= 2 & x_3 &= 2 & x_6 &= 1 \\x_2 &= 0 & x_4 &= 0 & x_5 &= 0 \\ \bar{z} &= 14\end{aligned}$$

et

$$\Delta_2 = 0 \quad \Delta_4 = 0 \quad \Delta_5 = -1$$

C'est une solution dégénérée. En appliquant le théorème précédent :

$$\begin{aligned} z &= 14 + 2x_4 - x_5 \\ x_1 &= 2 + \frac{7}{3}x_2 + x_4 + \frac{1}{3}x_5 \\ x_2 &= 2 - \frac{1}{3}x_2 - \frac{1}{3}x_5 \\ x_6 &= 1 - \frac{13}{3}x_2 - 4x_4 - \frac{1}{3}x_5 \end{aligned}$$

On trouve ici $\hat{\theta} = 3$. Une nouvelle solution de base est donc :

$$\mathcal{B}' = \{1, 3, 5\} \qquad \mathcal{N}' = \{2, 4, 6\}$$

d'où

$$\begin{array}{lll} x_1 = 3 & x_2 = 0 & \bar{z}' = 11 \\ x_3 = 1 & x_4 = 0 & \\ x_5 = 3 & x_6 = 0 & \end{array}$$

On recalcule les Δ_i et on constate qu'ils sont tous positifs ; ceci est notre critère d'arrêt.

6.2.2 Algorithme du simplexe pour la minimisation

L'algorithme 7 détaille l'algorithme du simplexe dans le cas d'un problème de minimalisation.

6.2.3 Méthode des tableaux

Soit le problème :

$$\max 4x_1 + 12x_2 + 3x_3$$

s.c.

$$\left\{ \begin{array}{ll} x_1 & \leq 1000 \\ x_2 & \leq 500 \\ x_3 & \leq 1500 \\ 3x_1 + 6x_2 + 2x_3 & \leq 6750 \\ x_1, x_2, x_3 & \geq 0 \end{array} \right.$$

Mettons-le d'abord sous forme standard. On a alors le problème équivalent :

$$\max 4x_1 + 12x_2 + 3x_3$$

Algorithme 7 Algorithme du simplexe pour la minimisation

déterminer une solution de base réalisable
calculer $B^{-1}N = (a_{ij})$ avec $i \in \mathcal{B}, j \in \mathcal{N}$
calculer les coûts réduits $\Delta_j = c_j - \sum_{i \in \mathcal{B}} c_i \bar{a}_{ij}$ ($j \in \mathcal{N}$)
si $\forall j \in \mathcal{N}, \Delta_j \geq 0$ **alors**
la solution considérée est optimale ; fin
sinon
 J ensemble des indices de \mathcal{N} t.q. $\Delta_j < 0$
si pour un indice $j \in J, \forall i \in \mathcal{B}, \bar{a}_{ij} < 0$ **alors**
optimum = $-\infty$; fin
sinon
choisir s t.q. $\Delta_s = \min_{j \in J} \Delta_j$ (critère d'entrée)
déterminer $r \in \mathcal{B}$ tel que $\frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{\substack{i \in \mathcal{B} \\ \bar{a}_{is} > 0}} \frac{\bar{b}_i}{\bar{a}_{is}}$
fin si
fin si
considérer la nouvelle base déduite de \mathcal{B} en remplaçant la colonne r par la colonne s .
reprendre à partir des calculs de $B^{-1}N$

s.c.

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

En posant le problème sous la forme $Ax = b$, la matrice A vaut donc :

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 3 & 6 & 2 & 0 & 0 & 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1000 \\ 500 \\ 1500 \\ 6750 \end{pmatrix}$$

Une solution de base réalisable est :

$$\begin{array}{ll} x_1 = 0 & x_4 = 1000 \\ x_2 = 0 & x_5 = 500 \\ x_3 = 0 & x_6 = 1500 \\ \bar{z} = 0 & x_7 = 6750 \end{array}$$

On trouve $\Delta_1 = 4, \Delta_2 = 12, \Delta_3 = 3$, et $\Delta_4 = \dots = \Delta_7 = 0$.

En appliquant l'algorithme du simplexe, on trouve $s = 2$ et $r = 2$.

On a alors :

$$c_i = \begin{pmatrix} 0 \\ 12 \\ 0 \\ 0 \end{pmatrix} \quad i \in \mathcal{B} = \begin{pmatrix} 4 \\ 2 \\ 6 \\ 7 \end{pmatrix} \quad A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 3 & 0 & 2 & 0 & -6 & 0 & 1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 1000 \\ 500 \\ 1500 \\ 3750 \end{pmatrix}$$

et $\Delta_1 = 4$, $\Delta_2 = 0$, $\Delta_3 = 3$, $\Delta_4 = 0$, $\Delta_5 = -12$, $\Delta_6 = \Delta_7 = 0$. Ce qui donne :

$$\begin{array}{ll} x_1 = 0 & x_4 = 1000 \\ x_3 = 0 & x_2 = 500 \\ x_5 = 0 & x_6 = 1500 \\ \bar{z} = 6000 & x_7 = 3750 \end{array}$$

Et une dernière itération en utilisant la ligne 1 comme pivot :

$$c_i = \begin{pmatrix} 4 \\ 12 \\ 0 \\ 0 \end{pmatrix} \quad i \in \mathcal{B} = \begin{pmatrix} 1 \\ 2 \\ 6 \\ 7 \end{pmatrix} \quad A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & -3 & -6 & 0 & 1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 1000 \\ 500 \\ 1500 \\ 750 \end{pmatrix}$$

et $\Delta_3 = 3$, $\Delta_4 = -4$, $\Delta_5 = -12$, $\Delta_1 = \Delta_2 = \Delta_6 = \Delta_7 = 0$.

Il faut ensuite encore quelques autres itérations. La solution optimale qu'on trouve ainsi est :

$$\begin{array}{ll} x_1 = 250 & x_4 = 750 \\ x_2 = 500 & x_5 = 0 \\ x_3 = 1500 & x_6 = 0 \\ \bar{z} = 11500 & x_7 = 0 \end{array}$$

Cet algorithme n'est pas fait pour être fait à la main, cependant.

6.3 Initialisation de l'algorithme du simplexe

Pour l'instant, on partait d'une base évidente pour appliquer l'algorithme du simplexe. Nous allons maintenant voir comment on peut trouver une base initiale (et de préférence déjà optimale).

On rappelle le problème $A\mathbf{x} = \mathbf{b}$ décomposé en variables de base et hors base :

$$\mathbf{x}^{\mathcal{B}} + B^{-1}N\mathbf{x}^{\mathcal{N}} = B^{-1}\mathbf{b}$$

On souhaite trouver une base réalisable initiale et l'écriture canonique des expressions :

- des variables de base en fonction des variables hors base
- de la fonction économique en fonction des variables hors base, i.e.

$$z = \bar{z} + \sum_{j \in \mathcal{N}} \Delta_j x_j$$

Si le problème est posé sous la forme $Ax \leq b$, on a vu qu'il y avait une base initiale composée des variables d'écart. En effet, chaque ligne i du problème s'écrit

$$\sum_{j=1}^n a_{ij} x_j + e_i = b_i$$

Cependant, tous les problèmes ne se posent pas toujours comme cela. Supposons que le problème soit déjà sous la forme standard, i.e. :

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

6.3.1 Problème auxiliaire : Méthode des 2 phases

Le problème auxiliaire PA s'écrit sous la forme :

$$(PA) \begin{cases} \min \varphi = \sum_{i=1}^m v_i \\ \sum_{j=1}^n a_{ij} x_j + v_i = b_i & i \in \llbracket 1, m \rrbracket \\ x_j \geq 0 & j \in \llbracket 1, n \rrbracket \\ v_i \geq 0 & i \in \llbracket 1, m \rrbracket \end{cases}$$

Les v_i sont appelées *variables artificielles*.

Propriété 14. Soit (\tilde{x}, \tilde{v}) une solution admissible de (PA) . \tilde{x} est une solution admissible de (P) ssi $\tilde{v} = 0$.

Propriété 15. Soit (x^*, v^*) une solution optimale de (PA) . Le problème (P) admet une solution réalisable ssi $v^* = 0$.

Le but est de résoudre le programme linéaire (PA) par l'algorithme du simplexe, et en utilisant la base réalisable initiale canonique telle que :

- les variables de base sont les $(v_i)_{i \in \llbracket 1, m \rrbracket} = (b_i)$,
- les variables hors base sont les $(x_j)_{j \in \llbracket 1, n \rrbracket} = 0$.

Théorème 16. Si (P) admet une solution réalisable, alors (PA) admet une base optimale qui est une base réalisable de (P) . On note φ^* la valeur optimale de (PA) .

Démonstration. Considérons une base optimale J de (PA) .

1. Si $\varphi^* > 0$, alors (P) n'a pas de solution.
2. Si $\varphi^* = 0$ et toutes les variables artificielles sont hors base, alors J est une base réalisable du problème initial.
3. Si $\varphi^* = 0$ mais certaines variables artificielles (bien que nulles) sont dans la base (dégénérescence) : une telle variable (d'indice notée r) s'écrit

$$v_r + \sum_{j \in \mathcal{N}} \bar{a}_{rj} x_j + \sum_{i \in \mathcal{N}} \bar{a}_{ri} v_i = \bar{b}_r = 0$$

(a) Si $\forall j \in \mathcal{N}, \bar{a}_{rj} = 0$: on obtient

$$v_r + \sum_{i \in \mathcal{N}} \bar{a}_{ri} v_i = 0$$

La ligne r est redondante ; on peut donc la supprimer.

(b) Si $\exists j \in \mathcal{N}, \bar{a}_{rj} \neq 0$: soit $s \in \mathcal{N}$ tel que $\bar{a}_{rs} \neq 0$. On fait rentrer la variable x_s et sortir v_r de la base. On itère jusqu'à ce que toutes les variables artificielles soient hors base. Ainsi, on revient au cas 2. \square

6.3.2 Exemples d'application

Exemple 14. Soit (P) le problème :

$$\left\{ \begin{array}{l} \max z = x_1 + x_2 + x_3 + x_4 \\ x_1 + 2x_2 + x_3 = 2 \\ x_1 + x_2 + 5x_3 = 12 \\ x_1 + 2x_2 + 6x_3 + x_4 = 13 \\ x_1, x_2, x_3, x_4 \geq 0 \end{array} \right.$$

Le problème auxiliaire s'écrit alors :

$$\left\{ \begin{array}{l} \min \varphi = v_1 + v_2 \\ x_1 + 2x_2 + x_3 + v_1 = 2 \\ x_1 + x_2 + 5x_3 + v_2 = 12 \\ x_1 + 2x_2 + 6x_3 + x_4 = 13 \\ x_1, \dots, x_4, v_1, v_2 \geq 0 \end{array} \right.$$

On a considéré x_4 comme variable artificielle car elle n'apparaît qu'une seule fois dans une seule des équations, et x_4 doit être positive. Les variables de base sont donc v_1, v_2 et x_4 . Ainsi, une solution de base réalisable initiale de (PA) est :

$$\begin{array}{ll} v_1 = 2 & x_1 = 0 \\ v_2 = 12 & x_2 = 0 \\ x_4 = 13 & x_3 = 0 \\ \bar{\varphi} = 14 & \end{array}$$

Appliquons maintenant la méthode des tableaux :

$$\begin{array}{cccccccc} \mathbf{c}_i & \mathcal{B} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{v}_1 & \mathbf{v}_2 \\ 1 & v_1 & 1 & 2 & 1 & 0 & 1 & 0 & 2 \\ 1 & v_2 & 1 & 1 & 5 & 0 & 0 & 1 & 12 \\ 0 & x_4 & 1 & 2 & 6 & 1 & 0 & 0 & 13 \\ & & & -2 & -3 & -6 & 0 & 0 & 0 \end{array}$$

En ce qui concerne la valeur des Δ_j , il faut utiliser la formule

$$\Delta_j = c_j - \sum_{i \in \mathcal{B}} c_i \bar{a}_{ij}$$

On a $\bar{\varphi} = 14$.

On choisit la ligne 1 et la colonne 3 comme pivots.

$$\begin{array}{cccccccc} \mathbf{c}_i & \mathcal{B} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{v}_1 & \mathbf{v}_2 \\ 0 & x_3 & 1 & 2 & 1 & 0 & 1 & 0 & 2 \\ 1 & v_2 & -4 & -9 & 0 & 0 & -5 & 1 & 2 \\ 0 & x_4 & -5 & -10 & 0 & 1 & -6 & 0 & 1 \\ & & & 4 & 9 & 0 & 0 & 6 & 0 \end{array}$$

avec $\bar{\varphi} = 2$. Comme tous les Δ_i sont positifs, nous avons là une solution optimale de (PA) :

$$\begin{array}{ll} x_3 = 2 & x_1 = 0 \\ x_4 = 1 & x_2 = 0 \\ v_2 = 2 & v_1 = 0 \\ \bar{\varphi} = 2 & \end{array}$$

Cependant, la valeur optimale $\bar{\varphi}$ est strictement positive, donc le problème initial (P) n'admet pas de solution.

Exemple 15. Modifions légèrement le problème (P) :

$$\left\{ \begin{array}{l} \max z = x_1 + x_2 + x_3 + x_4 \\ 2x_2 + x_3 = 2 \\ x_1 + x_2 + 5x_3 = 12 \\ x_1 + 2x_2 + 6x_3 + x_4 = 13 \\ x_1, x_2, x_3, x_4 \geq 0 \end{array} \right.$$

Le problème auxiliaire correspondant est :

$$\left\{ \begin{array}{l} \min \varphi = v_1 + v_2 \\ 2x_2 + x_3 + v_1 = 2 \\ x_1 + x_2 + 5x_3 + v_2 = 12 \\ x_1 + 2x_2 + 6x_3 + x_4 = 13 \\ x_1, \dots, x_4, v_1, v_2 \geq 0 \end{array} \right.$$

En appliquant la même méthode, on obtient le tableau

c_i	\mathcal{B}	x_1	x_2	x_3	x_4	v_1	v_2	
1	v_1	0	2	1	0	1	0	2
1	v_2	1	1	5	0	0	1	12
0	x_4	1	2	6	1	0	0	13
			-1	-3	-6	0	0	0

On choisit comme pivot la ligne 1, colonne 3. Au bout de trois itérations, le tableau optimum obtenu avec l'algorithme du simplexe est :

c_i	\mathcal{B}	x_1	x_2	x_3	x_4	v_1	v_2	
0	x_2	0	1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	1
1	v_2	0	0	$-\frac{1}{2}$	-1	$\frac{1}{2}$	1	0
0	x_1	1	0	5	1	-1	0	11
			0	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0

On a ainsi obtenu la solution optimale suivante pour (PA) :

$$\begin{array}{ll} x_1 = 11 & x_3 = 0 \\ x_2 = 1 & x_4 = 0 \\ v_2 = 0 & v_1 = 0 \\ \bar{\varphi} = 0 & \end{array}$$

Nous avons une solution optimale, mais nous avons encore une variable artificielle dans notre base. Nous allons alors enlever cette variable et en ajouter une autre non artificielle, en utilisant comme pivot la ligne 2, colonne 4. Cela donne :

$$\begin{array}{r}
 c_i \quad \mathcal{B} \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad v_1 \quad v_2 \\
 0 \quad x_2 \quad 0 \quad 1 \quad \frac{1}{2} \quad 0 \quad \frac{1}{2} \quad 0 \quad 1 \\
 1 \quad x_4 \quad 0 \quad 0 \quad \frac{1}{2} \quad 1 \quad -\frac{1}{2} \quad -1 \quad 0 \\
 0 \quad x_1 \quad 1 \quad 0 \quad \frac{3}{2} \quad 0 \quad -\frac{1}{2} \quad 1 \quad 11 \\
 \\ \\
 \prime \quad 0 \quad 0 \quad \frac{1}{2} \quad 1 \quad \frac{1}{2} \quad 0
 \end{array}$$

Ainsi, les variables de base sont x_1 , x_2 et x_4 . Nous avons désormais une base initiale afin d'appliquer l'algorithme du simplexe sur notre problème de maximisation (P). Le tableau sur lequel appliquer l'algorithme serait :

$$\begin{array}{r}
 c_i \quad \mathcal{B} \quad x_1 \quad x_2 \quad x_3 \quad x_4 \\
 1 \quad x_2 \quad 0 \quad 1 \quad \frac{1}{2} \quad 0 \quad 1 \\
 1 \quad x_4 \quad 0 \quad 0 \quad \frac{1}{2} \quad 1 \quad 0 \\
 1 \quad x_1 \quad 1 \quad 0 \quad \frac{3}{2} \quad 0 \quad 11 \\
 \\ \\
 \prime \quad 0 \quad 0 \quad -\frac{9}{2} \quad 0
 \end{array}$$

avec $\bar{z} = 12$. La solution est d'emblée optimale,¹ ce qui n'est pas toujours le cas ! Ainsi :

$$\begin{array}{ll}
 x_1 = 11 & x_3 = 0 \\
 x_2 = 1 & x_4 = 0 \\
 \bar{z} = 12 &
 \end{array}$$

6.3.3 Méthode des pénalités

Afin de résoudre (P), on résout le problème auxiliaire

$$\min \underbrace{\sum_{j=1}^n c_j x_j}_{\text{fonc. éco de } (P)} + M \sum_{i=1}^m v_i$$

avec M une *pénalité*. L'idée générale de la résolution reste cependant identique.

1. Pour un problème de maximisation, la condition nécessaire et suffisante d'optimalité est que les Δ_i soient négatifs ou nuls.

Chapitre 7

Processus aléatoires (ou stochastiques)

On se donne un ensemble de variables aléatoires $\{X(t), t \in \mathcal{T}\}$. On dit aussi que $X(t)$ est la *valeur du processus* à l'instant t .

L'ensemble \mathcal{T} est l'ensemble des *paramètres* (discrets ou continus). L'ensemble des valeurs du processus est appelé *espace des états* ; cet ensemble peut être discret ou continu.

Exemple 16 (File d'attente). On se donne un guichet où viennent se servir des clients. Chaque client est servi pendant un temps aléatoire. On note $X(t)$ ($t \in [0; +\infty[$) le nombre de clients dans le système d'attente à l'instant t . L'espace des états est \mathbb{N} .

7.1 Processus de Markov (sans mémoire)

Définition 19. Un processus $\{X(t), t \in \mathcal{T}\}$ est un *processus markovien* si, pour tout $n \in \mathbb{N}$, pour tout $t_1 < t_2 < \dots < t_n < t_{n+1}$,

$$\Pr(X(t_{n+1}) \in A | X(t_1), X(t_2), \dots, X(t_n)) = \Pr(X(t_{n+1}) \in A | X(t_n))$$

Autrement dit : la connaissance des événements antérieurs est inutile.

Définition 20 (Processus homogène). Si $\Pr(X(t+s) \in A | X(t))$ ne dépend pas de t , ce processus est dit *homogène*.

7.1.1 Processus de Poisson

Il s'agit d'un processus de Markov homogène dont l'espace des états est discret, mais l'ensemble des paramètres est continu.

On se donne un événement e qui se produit à intervalles aléatoires. Le processus est $\{X(t), t \in [0, +\infty[\}$, où $X(t)$ est le nombre d'événements qui se sont produits entre l'instant 0 et l'instant t .

Postulats

Soit h un intervalle de temps.

$$\begin{aligned} \Pr(X(t+h) - X(t) = 1) &= \lambda h + o(h) & \lambda > 0 \\ \Pr(X(t+h) - X(t) = 0) &= 1 - \lambda h + o(h) \end{aligned}$$

On note $P_m(t) = \Pr(X(t) = m)$, i.e. la probabilité que m événements se produisent entre les instants 0 et t . On souhaite calculer $P_m(t)$.

Calcul de $P_m(t)$

Soient $m \in \mathbf{N}$, $t \in \mathbf{R}_+$ et $dt > 0$.

Pour $m = 0$

$$\begin{aligned} P_0(t+dt) &= P_0(t)(1 - \lambda dt + o(dt)) \\ \Leftrightarrow \frac{P_0(t+dt) - P_0(t)}{dt} &= -\lambda P_0(t) + \frac{o(dt)}{dt} \end{aligned}$$

d'où

$$P_0'(t) = -\lambda P_0(t)$$

En résolvant cette équation différentielle, on obtient

$$P_0 = ce^{-\lambda t} \quad c \in \mathbf{R}$$

Lorsque $t = 0$, on a $P_0(0) = 1$, d'où $c = 1$. Donc

$$P_0(t) = e^{-\lambda t}$$

Pour $m \geq 1$ Ici, on peut écrire

$$P_m(t+dt) = P_{m-1}(t)\lambda dt + P_m(t)(1 - \lambda dt)$$

car on doit considérer le cas où un événement se produise entre t et $t+dt$, et celui où aucun événement ne se produit. On calcule de la même façon la dérivée, et on obtient :

$$P_m'(t) = \lambda P_{m-1}(t) - \lambda P_m(t)$$

Pour $m = 1$, on a :

$$P_1'(t) + \lambda P_1(t) = \lambda e^{-\lambda t}$$

La résolution de cette équation-là donne

$$P_1(t) = \lambda t e^{-\lambda t}$$

En appliquant le raisonnement analogue, on obtient de façon générale :

$$\boxed{P_m(t) = \frac{(\lambda t)^m e^{-\lambda t}}{m!}} \quad (7.1)$$

7.1.2 Processus de naissance

Ceci est une première généralisation du processus de Poisson. Soit un processus $\{X(t), t \in [0, +\infty[$ où $X(t)$ est l'effectif d'une population à l'instant t . On suppose que $X(0) = 0$. Les postulats deviennent les suivants :

$$\begin{aligned} Pr(X(t+h) - X(t) = 1 | X(t) = k) &= \lambda_k h + o(h) \\ Pr(X(t+h) - X(t) = 0 | X(t) = k) &= 1 - \lambda_k h + o(h) \end{aligned}$$

Pendant l'intervalle de temps h (qu'on fera tendre vers 0), il y a deux possibilités : une naissance ou zéro naissance.

Pour tout $(m, t) \in \mathbf{N} \times \mathbf{R}_+$, on note $P_m(t)$ la probabilité que l'effectif de la population soit égale à m à l'instant t .

Calcul de $P_m(t)$

Soit $t \geq 0$ et $dt > 0$. On a :

$$\begin{aligned} P_0(t+dt) &= P_0(t)(1 - \lambda_0 dt) \\ \Rightarrow P_0'(t) &= -\lambda_0 P_0(t) \end{aligned}$$

et

$$\begin{aligned} P_m(t+dt) &= P_{m-1}(t)\lambda_{m-1}dt + P_m(t)(1 - \lambda_m dt) \\ \Rightarrow P_m'(t) &= \lambda_{m-1}P_{m-1}(t) - \lambda_m P_m(t) \end{aligned}$$

Nous devons utiliser la transformée de Laplace (cf. annexe A). On a donc, pour $m \geq 1$:

$$pP_m(p) = \lambda_{m-1}P_{m-1}(p) - \lambda_m P_m(p)$$

$$\Rightarrow \boxed{P_m(p) = \lambda_{m-1} \frac{P_{m-1}(p)}{p + \lambda_m}}$$

et pour $m = 0$:

$$pP_0(p) - p = -\lambda_0 P_0(p)$$

$$\Rightarrow \boxed{P_0(p) = \frac{p}{p + \lambda_0}}$$

ce qui donne :

$$P_1(p) = \frac{\lambda_0}{p + \lambda_1} P_0(p)$$

$$P_2(p) = \frac{\lambda_1}{p + \lambda_2} P_1(p)$$

$$\vdots$$

$$P_m(p) = \frac{\lambda_{m-1}}{p + \lambda_m} P_{m-1}(p)$$

$$= \prod_{i=0}^{m-1} \frac{\lambda_i}{p + \lambda_{i+1}} P_0(p)$$

$$= \prod_{i=0}^{m-1} \frac{\lambda_i}{p + \lambda_{i+1}} p$$

On peut alors distinguer deux cas différents :

1. Tous les λ_i sont différents et non nuls :

$$\boxed{P_m(t) = \left(\prod_{i=0}^{m-1} \lambda_i \right) \sum_{k=0}^m \frac{e^{-\lambda_k t}}{\prod_{\substack{j=0 \\ j \neq k}}^m (\lambda_j - \lambda_k)}$$

2. Tous les λ_i sont égaux à une constante λ :

$$P_m(p) = \lambda^m \frac{p}{(p + \lambda)^{m+1}}$$

d'où

$$\boxed{P_m(t) = \frac{\lambda t e^{-\lambda t}}{m!}}$$

7.1.3 Processus de naissance et de mort

Ceci est une deuxième généralisation du processus de Poisson.

Soit $\{X(t), t \in [0, +\infty[\}$ le processus tel que $X(t)$ soit égale à l'effectif d'une population à l'instant t . Cette fois, on introduit la possibilité qu'il y ait une mort parmi la population pendant deux instants donnés. Les postulats ainsi réécrits deviennent :

$$\begin{aligned} \Pr(X(t+h) - X(t) = 1 | X(t) = k) &= \lambda_k h + o(h) && \text{Naissance} \\ \Pr(X(t+h) - X(t) = -1 | X(t) = k) &= \mu_k h + o(h) && \text{Mort} \\ \Pr(X(t+h) - X(t) = 0 | X(t) = k) &= 1 - (\lambda_k + \mu_k)h + o(h) && \text{Rien du tout} \end{aligned}$$

Les λ_i (resp. μ_i) sont appelés *taux de naissance* (resp. *taux de mort*). Tous les λ_i et μ_i sont strictement positifs, sauf μ_0 qui est nul.

Calcul de $P_m(t)$

Cette fois, $P_0(t)$ s'écrit

$$\begin{aligned} P_0(t+dt) &= P_0(t)(1 - \lambda_0 dt) + P_1(t)\mu_1 dt \\ \Rightarrow P'_0(t) &= -\lambda_0 P_0(t) + \mu_1 P_1(t) \end{aligned}$$

et pour $m \geq 1$:

$$P_m(t+dt) = P_{m-1}(t)\lambda_{m-1}dt + P_m(t)(1 - \lambda_{m-1}dt - \mu_{m-1}dt) + P_{m+1}(t)\mu_{m+1}dt$$

d'où

$$P'_m(t) = \lambda_{m-1}P_{m-1}(t) - (\lambda_m + \mu_m)P_m(t) + \mu_{m+1}P_{m+1}(t)$$

Afin d'alléger les notations, on pose

$$\Pi_n = \frac{\lambda_0 \dots \lambda_{n-1}}{\mu_1 \dots \mu_n} \quad \sum_{k=1}^{+\infty} \Pi_k \text{ converge}$$

On note également $\lim_{t \rightarrow +\infty} P_m(t) = P_m$.

Il s'agit en quelque sorte d'un régime permanent atteint après un certain régime transitoire ; on ne s'intéressera qu'aux calculs des probabilités en régime permanent (i.e. des P_m).

Résolvons le système suivant :

$$\begin{aligned} 0 &= -\lambda_0 P_0 + \mu_1 P_1 \\ 0 &= \lambda_{m-1} P_{m-1} - (\lambda_m + \mu_m) P_m + \mu_{m+1} P_{m+1} \text{ pour } m \geq 1 \end{aligned}$$

Soit $m \geq 1$. On a :

$$\begin{aligned}\lambda_m P_m - \mu_{m+1} P_{m+1} &= \lambda_{m-1} P_{m-1} - \mu_m P_m \\ &= \lambda_0 P_0 - \mu_1 P_1 \\ &= 0\end{aligned}$$

d'où

$$\begin{aligned}P_m &= \frac{\lambda_{m-1}}{\mu_m} P_{m-1} \\ &\vdots \\ P_1 &= \frac{\lambda_0}{\mu_1} P_0\end{aligned}$$

et donc

$$P_m = \frac{\lambda_0 \cdots \lambda_{m-1}}{\mu_1 \cdots \mu_m} P_0$$

Comme il s'agit d'une distribution de probabilités, on a

$$\sum_{m=0}^{+\infty} P_m = 1$$

et donc

$$P_m = \Pi_m \left(1 + \sum_{i=1}^{+\infty} \Pi_i \right)^{-1}$$

Considérons le cas particulier où $\lambda_0 = \lambda_1 = \cdots = \lambda$ et $\mu_0 = \mu_1 = \cdots = \mu$. Alors, on a :

$$P_m = \underbrace{\left(\frac{\lambda}{\mu} \right)^m}_{\Pi_m} P_0$$

Afin que le régime permanent puisse s'établir, il faut que $\frac{\lambda}{\mu} < 1$, sinon $\sum \Pi_n$ diverge.

Ainsi,

$$\begin{aligned}\sum_{m=0}^{+\infty} P_m = 1 &\Rightarrow P_0 \sum_{m=0}^{+\infty} \left(\frac{\lambda}{\mu} \right)^m = 1 \\ &\Rightarrow P_0 = \left(\sum_{m=0}^{+\infty} \left(\frac{\lambda}{\mu} \right)^m \right)^{-1} \\ &\Rightarrow P_0 = 1 - \frac{\lambda}{\mu}\end{aligned}$$

Exemple d'application

En utilisant le modèle démographique de *croissance linéaire avec immigration*, on a ici :

- Taux de naissance : $\lambda_n = n\lambda + a$
- Taux de mort : $\mu_n = n\mu$

On note toujours $X(t)$ l'effectif de la population à l'instant t , et on souhaite toujours calculer $P_n(t) = \Pr(X(t) = n)$.

$$P_0(t + dt) = P_0(t)(1 - a dt) + P_1(t)\mu dt$$

et donc

$$\boxed{P_0'(t) = -aP_0(t) + \mu P_1(t)}$$

Pour $n \geq 1$:

$$P_n(t + dt) = P_{n-1}(t)((n-1)\lambda + a)dt + P_n(t)(1 - (n\lambda + a + n\mu)dt) + P_{n+1}(t)(n+1)\mu dt$$

ce qui donne

$$P_n'(t) = ((n-1)\lambda + a)P_{n-1}(t) - (n\lambda + a + n\mu)P_n(t) + (n+1)\mu P_{n+1}(t)$$

On note $M(t)$ l'espérance mathématique de l'effectif à l'instant t . Ainsi :

$$M(t) = \sum_{n=0}^{+\infty} nP_n(t) \qquad M'(t) = \sum_{n=0}^{+\infty} nP_n'(t)$$

On constate que $M'(t) = a + (\lambda + \mu)M(t)$, d'où l'équation différentielle

$$\boxed{M'(t) + (\mu - \lambda)M(t) = a}$$

Résolvons cette équation. Si on note i l'effectif à $t = 0$, on a :

$$M(t) = \begin{cases} at + i & \text{si } \lambda = \mu \\ \left(i - \frac{a}{\mu - \lambda}\right) \exp((\lambda - \mu)t) + \frac{a}{\mu - \lambda} & \text{sinon} \end{cases}$$

Si $\lambda > \mu$, alors $\lim_{t \rightarrow +\infty} M(t) = +\infty$. Si $\lambda < \mu$, alors $M(t)$ se stabilise autour de la grandeur $\frac{a}{\mu - \lambda}$.

7.2 Chaînes de Markov

Il s'agit d'un processus de Markov dont l'ensemble des indices est discret (contrairement aux processus de Markov), mais l'espace des états est toujours discret.

On note une telle chaîne $\{X_n; n = 0, 1, 2, \dots\}$.

Propriété 17. *L'état actuel ne dépend que de l'état précédent ; ce processus est sans mémoire. Autrement dit : Soient $n \geq 1$ et $(j_k)_{k \in \llbracket 0, n \rrbracket} \in S^{n+1}$ valeurs dans l'espace des états. Alors*

$$\Pr(X_n = j_n | X_i = j_i, i \in \llbracket 0, n-1 \rrbracket) = \Pr(X_n = j_n | X_{n-1} = j_{n-1})$$

Par ailleurs, la grandeur $\Pr(X_n = j | X_{n-1} = i)$ ne dépend pas de n et est notée p_{ij} . On appelle probabilité de transition cette grandeur.

Exemple 17 (Ruine du joueur). On se donne 2 joueurs A et B qui commencent avec les capitaux respectifs a et $b \in \mathbb{E}$. À chaque pas, on tire à pile ou face. Si pile, B donne $1 \in$ à A . Si face sort, A donne $1 \in$ à B . Le jeu s'arrête lorsqu'un des deux joueurs est ruiné.

On pose $s = a + b$. Afin de caractériser les états de la chaîne, on considérera le capital du joueur A . L'espace des états est donc $\llbracket 0, s \rrbracket$.

Soit $(i, j) \in \llbracket 0, s \rrbracket^2$. Les probabilités de transition s'expriment de la façon suivante :

$$p_{ij} = \begin{cases} p \text{ (pile)} & \text{si } j = i + 1, i \neq 0 \\ q \text{ (face)} & \text{si } j = i - 1, i \neq s \\ 1 & \text{si } i = 0 \text{ ou } i = s \\ 0 & \text{dans les autres cas} \end{cases}$$

7.2.1 Matrice associée à une chaîne de Markov finie

Soit une chaîne de Markov avec r états. La matrice $(p_{ij})_{(i,j) \in \llbracket 1, r \rrbracket^2}$ est une matrice *stochastique*, i.e.

$$\forall i \in \llbracket 1, r \rrbracket, \sum_{j=1}^r p_{ij} = 1$$

Dans l'exemple 17, cette matrice est de la forme :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & q & 0 & p & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & q & 0 & p \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

avec $p + q = 1$.

Définition 21. Soit $P = (p_{ij})$ une matrice carrée d'ordre r . Cette matrice est dite *stochastique* si :

- $\forall (i, j) \in \llbracket 1, r \rrbracket^2, p_{ij} \in [0; 1]$;
- $\forall i \in \llbracket 1, r \rrbracket, \sum_{j=1}^r p_{ij} = 1$.

On pose $Q(n)$ une distribution de la probabilité des états à la transition n telle que

$$Q(n) = (q_1(n), \dots, q_i(n), \dots, q_r(n))$$

où $q_i(n)$ est la probabilité d'être dans l'état i à la transition n . Ainsi, on a :

$$q_i(n+1) = \sum_{k=1}^r q_k(n) p_{ki}$$

Si on note P la matrice des probabilités de transition, on a :

$$\begin{aligned} Q(n+1) &= Q(n)P \\ Q(n) &= Q(0)P^n \end{aligned}$$

7.2.2 Graphe associé à une chaîne

Le graphe associé à une chaîne est le graphe $G = (X, U)$ tel que :

- les sommets sont les états de la chaîne ;
- l'arc (i, j) appartient à U ssi $p_{ij} > 0$.

Exemple 18. Prenons la matrice P suivante :

$$P = \begin{pmatrix} \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

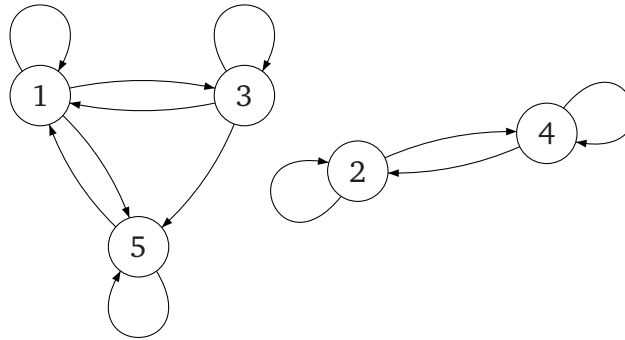


FIGURE 7.1 – Chaîne de Markov

Le graphe associé est celui de la figure 7.1. On constate ici que celui-ci comporte deux composantes connexes.

Exemple 19. Dans le problème du ruine du joueur (cf. exemple 17), le graphe associé est celui de la figure 7.2.

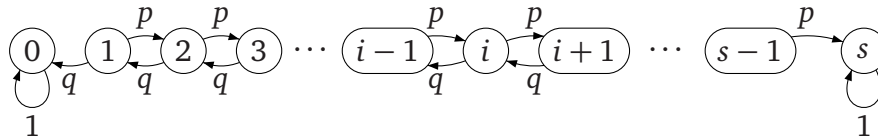


FIGURE 7.2 – Graphe associé à l'exemple 17

7.2.3 Classification des états

Définition 22. Soient E_1, E_2, \dots, E_r des états. On définit une relation d'équivalence R telle que :

$$E_i R E_j \iff \exists(m, n), p_{ij}^{(n)} > 0 \text{ et } p_{ji}^{(m)} > 0$$

où $p_{ij}^{(n)}$ est la probabilité de passer de i à j en n transitions. Autrement dit :

$$E_i R E_j \iff \text{il existe un chemin de } i \text{ à } j \text{ et de } j \text{ à } i$$

Propriété 18. Le terme $p_{ij}^{(n)}$ est le terme (i, j) de la matrice P^n .

Définition 23 (États communicants). E_i et E_j sont deux états communicants ssi $E_i R E_j$. On peut alors partitionner en classes d'équivalences appelées *classes d'états communicants*. Il s'agit alors des composantes fortement connexes du graphe associé.

Définition 24 (Ensemble fermé d'états). C est un ensemble d'états fermé si $\forall E_i \in C, \forall E_j \notin C, p_{ij} = 0$.

Si C est un singleton, son unique élément est appelé *état absorbant*.

Définition 25. Une *chaîne irréductible* est une chaîne qui ne contient pas de sous-ensembles fermé d'états.

Propriété 19. Dans une chaîne irréductible, tous les états communiquent.

Démonstration. Supposons que E_i et E_j ne communiquent pas (par exemple, on ne peut atteindre E_j à partir de E_i). Soit C l'ensemble de tous les états que l'on peut atteindre à partir de E_i . Par définition, $E_j \notin C$, donc la chaîne n'est pas irréductible. \square

Propriété 20. Une chaîne est irréductible ssi le graphe associé est fortement connexe.

Définition 26. On note $f_{ij}^{(n)}$ la *probabilité de premier passage*, i.e. la probabilité de passer en j pour la première fois au bout de n transitions en partant de i .

On pose également

$$F_{ij}^{(n)} = \sum_{k=1}^n f_{ij}^{(k)}$$

la probabilité, en partant de l'état i , de passer pour la première fois en j avant la transition $n + 1$. Enfin, on pose

$$F_{ij} = \lim_{n \rightarrow +\infty} F_{ij}^{(n)}$$

Définition 27. E_i est un *état transitoire* ssi $F_{ii} < 1$. E_i est dit *persistant* si $F_{ii} = 1$.

État périodique (ou persistant)

Pour chaque état E_i , on calcule $d_i = \text{pgcd}\{n \mid p_{ii}^{(n)} > 0\}$. Il s'agit, en fait, du PGCD de la longueur des circuits passant par l'état i .

Définition 28. L'état E_i est :

- *périodique* de période d_i si $d_i > 1$;

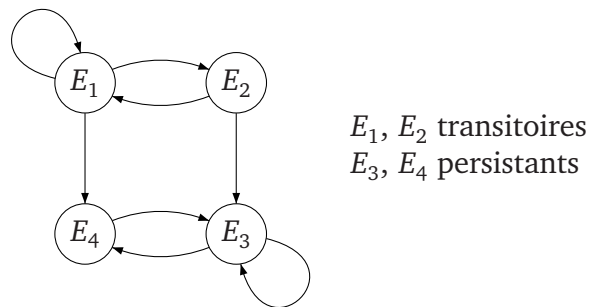


FIGURE 7.3 – États transitoires et permanents

– apériodique si $d_i = 1$.

Théorème 21. *Tous les états d'une classe d'états communicants ont la même période.*

Une condition suffisante de périodicité d'une classe d'états communicants est la présence d'une boucle sur un des états de la classe.

7.2.4 Distribution limite d'une chaîne de Markov finie

On rappelle qu'on a $Q(n+1) = Q(n)P$, donc $Q(n) = Q(0)P^n$. Comment se comporte $Q(n)$ lorsque n tend vers $+\infty$?

Définition 29. Une chaîne est *régulière* si elle admet une distribution limite indépendante de l'état initial.

On note $Q^* = \lim_{n \rightarrow +\infty} Q(n)$.

Théorème 22 (Condition nécessaire et suffisante de régularité). *Une chaîne est régulière ssi le graphe associé possède une seule classe finale d'états communicants non périodique.*

Calcul de la distribution limite

Une idée pour calculer Q^* est d'utiliser la formule

$$Q^* = Q(0)P^* \text{ où } P^* = \lim_{n \rightarrow +\infty} P^n$$

Une autre méthode est d'utiliser le fait que si Q^* est la distribution limite, alors $Q^* = Q^*P$. Cela donne un système d'équations linéaires. En outre, il faut que $\sum_{i=1}^r q_i^* = 1$.

Exemple 20. Dans le graphe de la figure 20, il y a une classe finale non périodique, donc la chaîne est régulière.

La matrice P de transition est donnée par :

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

On doit alors résoudre le système d'équations

$$\left\{ \begin{array}{l} (q_1^*, \dots, q_4^*) = (q_1^*, \dots, q_4^*)P \\ q_1^* + q_2^* + q_3^* + q_4^* = 1 \\ q_1^* = \frac{1}{2}q_1^* + \frac{1}{2}q_3^* \\ q_2^* = \frac{1}{2}q_1^* + \frac{1}{2}q_3^* \\ q_3^* = \frac{2}{3}q_2^* + q_4^* \\ q_4^* = \frac{1}{3}q_2^* \end{array} \right.$$

$$\Rightarrow 3q_1^* + \frac{1}{3}q_1^* = 1$$

$$\Rightarrow q_1^* = q_2^* = q_3^* = \frac{3}{10}; q_4^* = \frac{1}{10}$$

La probabilité de se retrouver dans l'état 1, 2 ou 3 après un temps infini vaut $\frac{3}{10}$, et celui de se retrouver dans l'état 4 vaut $\frac{1}{10}$.

Chapitre 8

Phénomènes d'attente

On se donne un *guichet* (ou *station*) qui sert une file de *clients* (aussi appelés *unités*). Cette file est illimitée, et on applique la discipline « premier arrivé, premier servi ».

Les arrivées sont poissonniennes, i.e. la probabilité d'arrivée de k clients pendant une durée t vaut

$$\frac{e^{-\lambda t} (\lambda t)^k}{k!}$$

où λ est le *taux d'arrivée*.

Les services sont exponentiels, i.e. la durée du service suit une loi de densité $\mu e^{-\mu t}$, où μ est le *taux de service*.

On souhaite calculer $P_n(t)$, la probabilité d'avoir n clients dans le système à l'instant t .

Soit θ la variable aléatoire de l'intervalle de temps entre deux arrivées consécutives. Alors :

$$\begin{aligned} F(t) &= \Pr(\theta \leq t) = 1 - P_0(t) \\ f(t) &= \lambda e^{-\lambda t} = 1 - e^{-\lambda t} \end{aligned}$$

8.1 Système à un guichet

Soit $X(t)$ la variable aléatoire du nombre de clients dans le système à t . Il s'agit d'un processus de naissance et de mort, avec un taux de naissance (i.e. un taux d'arrivée) de λ et un taux de mort (i.e. un taux de service) de μ .

Un régime permanent existe si $\frac{\lambda}{\mu} < 1$; sinon, les clients deviendront trop nombreux, et $X(t)$ tendra vers $+\infty$. En régime permanent, on a vu que :

$$P_n = \lim_{t \rightarrow +\infty} P_n(t) = \frac{\lambda_0 \cdots \lambda_{n-1}}{\mu_1 \cdots \mu_n} P_0$$

Ici, tous les λ_i sont égaux, ainsi que tous les μ_i . On rappelle alors qu'on trouvait

$$P_0 = 1 - \frac{\lambda}{\mu}$$

et donc

$$P_n = \left(\frac{\lambda}{\mu}\right)^n \left(1 - \frac{\lambda}{\mu}\right)$$

8.1.1 Nombre moyen de clients dans le système en régime permanent

Calculons le nombre moyen de clients dans le système en régime permanent. En d'autres termes : l'espérance de $X(t)$ quand $t \rightarrow +\infty$:

$$\begin{aligned} \bar{n} &= \sum_{n=0}^{+\infty} n P_n \\ &= \sum_{n=0}^{+\infty} n \left(\frac{\lambda}{\mu}\right)^n \left(1 - \frac{\lambda}{\mu}\right) \\ &= \left(1 - \frac{\lambda}{\mu}\right) \frac{\lambda}{\mu} \sum_{n=1}^{+\infty} n \left(\frac{\lambda}{\mu}\right)^{n-1} \\ &= \left(1 - \frac{\lambda}{\mu}\right) \frac{\lambda}{\mu} \left[\sum_{n=0}^{+\infty} \left(\frac{\lambda}{\mu}\right)^n \right]' \\ &= \left(1 - \frac{\lambda}{\mu}\right) \frac{\lambda}{\mu} \left[\frac{1}{1 - \frac{\lambda}{\mu}} \right]' \\ &= \left(1 - \frac{\lambda}{\mu}\right) \frac{\lambda}{\mu} \frac{1}{\left(1 - \frac{\lambda}{\mu}\right)^2} \end{aligned}$$

et donc

$$\bar{n} = \frac{\lambda}{\mu - \lambda}$$

Exemple 21. Application numérique : $\lambda = 18$ clients·h⁻¹, avec une durée

moyenne du service de 2 minutes (ce qui donne $\mu = 30 \text{ h}^{-1}$). On a alors :

$$\bar{n} = \frac{18}{12} = 1,5 \text{ client}$$

$$P_0 = 1 - \frac{18}{30} = \frac{2}{5}$$

8.1.2 Nombre moyen de clients dans la file

Cette grandeur est exprimée par :

$$\begin{aligned} \bar{v} &= \sum_{n=2}^{+\infty} (n-1)P_n \\ &= \sum_{n=2}^{+\infty} nP_n - \sum_{n=2}^{+\infty} P_n \\ &= \bar{n} - 1 + P_0 \end{aligned}$$

8.2 Système à plusieurs guichets identiques

On se donne cette fois S guichets identiques avec le même taux de service μ . On prend une file illimitée, commune aux S guichets, avec la même discipline, et les mêmes lois que précédemment. On peut ainsi servir S clients simultanément.

Il s'agit toujours d'un processus de naissance et de mort, avec un taux de naissance de λ , mais cette fois un taux de mort différent :

$$\mu_n = \begin{cases} S\mu & \text{si } n \geq S \\ n\mu & \text{si } n < S \end{cases}$$

Ainsi, en reprenant les calculs vus précédemment :

$$\begin{aligned} P_n &= \frac{\lambda^n}{\mu_1 \mu_2 \cdots \mu_n} P_0 \\ &= \begin{cases} \frac{\lambda^n}{n! \mu^n} P_0 & \text{si } 1 \leq n \leq S \\ \frac{\lambda^n}{S! S^{n-S} \mu^n} P_0 & \text{si } n \geq S \end{cases} \end{aligned}$$

Sachant que $\sum_{n=0}^{+\infty} P_n = 1$, on a :

$$P_0 = \left(\sum_{n=0}^{S-1} \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n + \sum_{n=S}^{+\infty} \left(\frac{\lambda}{\mu} \right)^n \frac{1}{S! S^{n-S}} \right)^{-1}$$

En outre :

$$\left(\frac{\lambda}{\mu}\right)^S \frac{1}{S!} \sum_{n=S}^{+\infty} \left(\frac{\lambda}{S\mu}\right)^{n-S} = \left(\frac{\lambda}{\mu}\right)^S \frac{1}{S!} \frac{1}{1 - \frac{\lambda}{S\mu}}$$

d'où

$$P_0 = \left(\sum_{n=0}^{S-1} \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^S}{S! \left(1 - \frac{\lambda}{S\mu}\right)} \right)^{-1}$$

Un régime permanent est possible à condition que $\frac{\lambda}{S\mu} < 1$.

On peut alors recalculer la densité f de la variable aléatoire du temps d'attente. On obtient alors la probabilité d'attendre plus que θ pour être servi.

Combien faut-il alors ouvrir de guichets pour que la probabilité d'attendre plus que θ soit inférieure à une certaine valeur P_{max} ?

Chapitre 9

Fiabilité des équipements

9.1 Généralités

On se donne des équipements pour lesquels on a observé une certaine fiabilité, et on souhaite optimiser l'utilisation de ces équipements afin de minimiser les problèmes d'exploitation dus à des pannes.

Définition 30. Soit T une variable aléatoire. On appelle *fiabilité* d'un équipement la grandeur

$$v(t) = \Pr(T > t)$$

i.e. la probabilité que l'équipement fonctionne encore correctement au bout d'un temps t .

On appelle *fonction de mortalité* (notée $i(t)$) la densité de T . Ainsi :

$$\Pr(t < T < t + dt) = i(t)dt$$

et

$$i(t) = \frac{d(1 - v(t))}{dt} = -v'(t)$$

Le *taux d'avarie*, noté $\lambda(t)$, est donné par la formule

$$\lambda(t)dt = \Pr(t < T < t + dt | T > t) = \frac{i(t)dt}{v(t)} = -\frac{v'(t)}{v(t)}$$

La *durée de vie moyenne* d'un équipement est donnée par :

$$\begin{aligned}\bar{t} &= \int_0^{+\infty} ti(t)dt \\ &= [-tv(t)]_0^{+\infty} - \int_0^{+\infty} v(t)dt\end{aligned}$$

Comme $tv(t)$ tend vers 0 si $t \rightarrow +\infty$, on a :

$$\bar{t} = \int_0^{+\infty} v(t) dt$$

9.2 Formes analytiques de la fiabilité et du taux d'avarie

On souhaite obtenir des formules explicites pour $v(t)$ et pour $\lambda(t)$. Les événements de panne sont supposés aléatoires et suivant une loi de Poisson. Ainsi :

$$\Pr(k \text{ événements pendant } t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

On suppose que le premier événement survenu entraîne la panne de l'équipement¹. Ainsi, la fiabilité s'exprime par :

$$\begin{aligned} v(t) &= \Pr(T > t) \\ &= \Pr(0 \text{ événement pendant } t) \\ &= e^{-\lambda t} \end{aligned}$$

et le taux d'avarie vaut :

$$\lambda(t) = -\frac{v'(t)}{v(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda$$

9.3 Généralisation de la loi exponentielle : loi Erlang-k

On reprend le modèle précédent, sauf que contrairement au celui-ci, k événements sont nécessaires pour provoquer une panne. Ainsi :

$$\begin{aligned} v(t) &= \Pr(T > t) = \sum_{n=0}^{k-1} \frac{e^{-\lambda t} (\lambda t)^n}{n!} \\ i(t) &= \lambda e^{-\lambda t} \frac{(\lambda t)^{k-1}}{(k-1)!} \end{aligned}$$

1. Ce modèle n'est plus souvent utilisé en pratique, mais il s'avère pertinent dans le cas où le matériel n'est pas sujet à l'usure.

9.4 Matériel d'usure

Soit p une fonction croissante. Alors :

$$\begin{aligned}\lambda(t) &= \lambda p(t) \\ -\frac{v'(t)}{v(t)} &= \lambda p(t)\end{aligned}$$

d'où

$$\log v(t) = -\int_0^t \lambda p(u) du$$

et donc

$$v(t) = \exp\left(-\lambda \int_0^t p(u) du\right)$$

Exemple 22 (Loi de Weibull). Ici, le taux d'avarie vaut

$$\lambda(t) = \alpha \lambda^\alpha t^{\alpha-1}$$

d'où

$$\begin{aligned}v(t) &= \exp(-(\lambda t)^\alpha) \\ i(t) &= \alpha \lambda^\alpha t^{\alpha-1} e^{-(\lambda t)^\alpha}\end{aligned}$$

9.5 Fiabilité des systèmes

On se donne un système à n composants, et on voudrait connaître la fiabilité du système, connaissant la fiabilité de chaque composant.

9.5.1 Système en série

Le système ne fonctionne que si tous ses composants fonctionnent. La panne de l'une d'entre elles entraîne la panne du système (ex. : RAID 0).

Autrement dit : de manière générale, le système ne fonctionne que s'il existe une chaîne entre E et S .

Pour chaque composant i , on connaît sa durée de vie T_i , ainsi que sa fiabilité $v_i(t)$. On souhaite calculer T et $v(t)$, qui sont la durée de vie et la fiabilité du système entier.

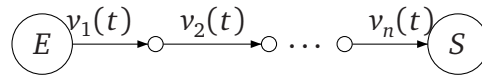


FIGURE 9.1 – Système en série

Or :

$$\Pr(T > t) = \Pr(T_1 > t) \cdot \Pr(T_2 > t) \cdot \dots \cdot \Pr(T_n > t)$$

donc

$$v(t) = \prod_{i=1}^n v_i(t)$$

9.5.2 Système en parallèle

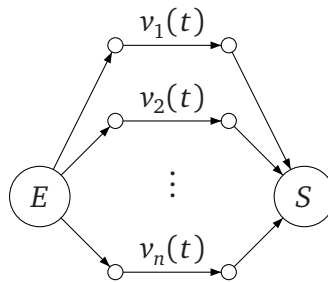


FIGURE 9.2 – Système en parallèle

Ici, le système contient plusieurs éléments redondants : le système ne tombe en panne que si *tous* ses composants tombent en panne (ex. : RAID 1). Cela se traduit de manière identique :

$$\Pr(T < t) = \prod_{i=1}^n (1 - v_i(t))$$

d'où

$$v(t) = 1 - \prod_{i=1}^n (1 - v_i(t))$$

Exemple 23. On peut alors imaginer des systèmes en série mis en parallèle, ou des système en parallèle mis en série. Lesquels sont les plus fiables ?

9.6 Probabilité de consommation

On appelle *consommation* le nombre de remplacements d'un matériel pendant un temps t . Autrement dit :

$$\begin{aligned} P_m(t) &= \Pr(\text{effectuer } m \text{ remplacements pendant } t) \\ &= \int_0^t i(u) \cdot P_{m-1}(t-u) du \end{aligned}$$

En appliquant la transformée de Laplace (et en notant $I(p) = Li(t)$) :

$$LP_m(t) = \frac{P_{m-1}(p)I(p)}{p}$$

En appliquant ainsi cette formule récurrente, on a :

$$\begin{aligned} P_0(p) &= V(p) \\ P_1(p) &= \frac{1}{p} P_0(p) I(p) \\ &\vdots \\ P_m(p) &= \frac{1}{p} P_{m-1}(p) I(p) \end{aligned}$$

d'où

$$P_m(p) = \frac{1}{p^m} I(p)^m V(p)$$

Il ne reste plus qu'à calculer $V(p)$. Or, sachant que

$$v(t) = 1 - \int_0^t i(u) du$$

on obtient alors

$$V(p) = 1 - \frac{I(p)}{p}$$

et donc

$$\begin{aligned} P_m(p) &= \frac{1}{p^m} (p(1 - V(p)))^m V(p) \\ &= (1 - V(p))^m V(p) \end{aligned}$$

Refaire la transformée inverse est fastidieux. Cependant, en appliquant le cas particulier où $v(t) = e^{-\lambda t}$, on a

$$V(p) = \frac{p}{p + \lambda}$$

et donc

$$\begin{aligned} P_m(p) &= \left(1 - \frac{p}{p + \lambda}\right)^m \frac{p}{p + \lambda} \\ &= \lambda^m \frac{p}{(p + \lambda)^{m+1}} \end{aligned}$$

La transformée inverse donne alors

$$P_m(t) = \frac{e^{-\lambda t} (\lambda t)^m}{m!}$$

9.7 Entretien préventif

Une première stratégie est de remplacer le matériel dès qu'il tombe en panne, et seulement lorsqu'il tombe en panne.

Une deuxième stratégie, cependant, consiste à faire de l'*entretien préventif*, en remplaçant systématiquement le matériel au bout d'un certain temps θ , ou lorsqu'il tombe en panne.

On note P le coût dû à la panne, et p le coût d'un matériel neuf. À chaque panne, le coût est de $P + p$; à chaque remplacement préventif, cela ne coûte que p .

9.7.1 Calcul du coût moyen

Stratégie 1 Le coût moyen vaut

$$\frac{p + P}{\bar{t}}$$

Stratégie 2 En notant \bar{t}_θ la durée de vie moyenne du composant sachant qu'il est remplacé tous les θ , on obtient :

$$\frac{p + P(1 - v(\theta))}{\bar{t}_\theta}$$

avec

$$\begin{aligned}\bar{t} &= \int_0^{+\infty} v(t) dt \\ \bar{t}_\theta &= \int_0^\theta ti(t) dt + \int_\theta^{+\infty} \theta i(t) dt \\ &= \int_0^\theta v(t) dt\end{aligned}$$

9.7.2 Conclusion

L'entretien préventif est intéressant s'il existe θ tel que :

$$\frac{p + P}{\int_0^{+\infty} v(t) dt} > \frac{p + P(1 - v(\theta))}{\int_0^{+\infty} v(t) dt}$$

Exemple 24. Dans le cas d'une fiabilité exponentielle $v(t) = e^{-\lambda t}$:

$$\bar{t} = \frac{1}{\lambda} \qquad \bar{t}_\theta = \frac{1}{\lambda} - \frac{e^{-\lambda\theta}}{\lambda}$$

Alors, l'entretien préventif est intéressant si

$$\begin{aligned}\frac{p + P(1 - e^{-\lambda\theta})}{\frac{1}{\lambda} - \frac{e^{-\lambda\theta}}{\lambda}} - \frac{p + P}{\frac{1}{\lambda}} < 0 \\ \iff pe^{-\lambda\theta} < 0\end{aligned}$$

ce qui est impossible. Ce résultat est cependant attendu, car dans ce cas précis, il n'y a pas d'usure...

Annexe A

Transformée de Laplace

Soit f une fonction définie sur \mathbf{R}_+ . On note $Lf(x)$ ou F la transformée de Laplace de f , qui est une fonction de \mathbf{C} et qui s'exprime par :

$$Lf(x): p \mapsto p \int_0^{+\infty} e^{-px} f(x) dx$$

Transformées usuelles

$$La = a$$

$$Lax = \frac{a}{p}$$

$$Le^{-ax} = \frac{p}{p+a}$$

$$L \frac{df}{dx}(x) = pLf(x) - pf(0)$$

$$L \int_0^x f(u) du = \frac{1}{p} Lf(x)$$

$$Le^{ax} f(x) = \frac{p}{p-a}$$

$$L \frac{x^{n-1}}{(n-1)!} e^{ax} = \frac{p}{(p+a)^n}$$

$$L \int_0^x f_1(x) f_2(x-u) du = \frac{1}{p} F_1(p) F_2(p)$$