

INFO-F-310 - Algorithmique 3 et Recherche Opérationnelle

Yves De Smet

Bernard Fortz

2015-2016

Table des matières

I	Introduction	3
1	Aide à la décision et modèles mathématiques	3
2	Quelques exemples de modèles mathématiques	4
3	Tour d'horizon des techniques de recherche opérationnelle	5
II	Programmation linéaire	6
4	Définition, exemples et méthode de résolution	6
4.1	Notions de bases	6
4.2	Exemples de modèles linéaires	6
4.3	Forme standard et forme canonique d'un programme linéaire	8
4.4	Résolution de programmes linéaires	10
4.4.1	Résolution graphique	10
4.4.2	La méthode du simplexe	12
4.4.3	La méthode des deux phases	16
4.4.4	Cas particuliers	17
5	Dualité	20
5.1	Le problème dual	20
5.2	Relations primal/dual	21
5.3	Interprétation économique de la dualité	22
5.4	Méthode simplexe duale	23
6	Solveurs et langages de modélisation	25
III	Le problème de transport	30
7	Définition et exemples	30
8	Algorithme pour le problème de transport	31
9	Le problème de transbordement	34
IV	Le flot maximum	36
10	Introduction	36
11	Formulation	36

12 Coupe minimum	36
13 Algorithme de Ford-Fulkerson	36
V Flots de coût minimum	39
14 Définitions et exemples	39
15 Algorithme de chemins augmentants	39
16 Extensions	44
VI Couplages	46
17 Introduction	46
18 Affectations bipartites	47
19 Le problème des mariages stables	52
VII La programmation dynamique	54
20 Retour aux plus courts chemins	54
21 Le problème de sac-à-dos	54
22 Détermination de la taille des lots	56

Première partie

Introduction

1 Aide à la décision et modèles mathématiques

Un premier problème

Exemple 1 (Achat de billets d'avion).

- Un homme d'affaires doit effectuer 5 voyages entre Fayetteville (FYV) à Denver (DEN), en partant le lundi de FYV et revenant le mercredi de DEN à FYV.
- Billet aller-retour : \$400.
- Réduction de 20 % si un weekend est inclus.
- Aller simple : 75 % du prix aller-retour.

Question

Comment acheter les billets pour les 5 semaines (à prix minimum) ?

Aide à la décision

Problème d'aide à la décision

1. Quelles sont les alternatives possibles ?
2. Quelles sont les restrictions à cette décision ?
3. Quel est l'objectif utilisé pour évaluer les alternatives ?

Restrictions

FYV-DEN le lundi et DEN-FYV le mercredi de la même semaine.

Evaluation des alternatives

Alternatives

- Acheter 5 FYV-DEN-FYV normaux. $5 \times \$400 = \2000
- Acheter un FYV-DEN, 4 DEN-FYV-DEN comprenant un weekend et un DEN-FYV. $0.75 \times \$400 + 4 \times 0.8 \times \$400 + 0.75 \times \$400 = \1880
- Acheter un FYV-DEN-FYV pour le lundi de la première semaine et le mercredi de la dernière semaine, et 4 DEN-FYV-DEN comprenant un weekend pour les autres voyages. $5 \times 0.8 \times \$400 = \1600

La troisième alternative est la meilleure.

Modèle de recherche opérationnelle

Ingrédients principaux

- *Alternatives* (variables, inconnues du problème).
- *Restrictions* (contraintes).
- *Fonction objectif* à optimiser (minimiser ou maximiser).

Définition 1 (Solution admissible). Une *solution admissible* est un ensemble de valeurs données aux variables qui satisfait toutes les contraintes.

Définition 2 (Solution optimale). Une *solution optimale* est une solution admissible qui optimise la fonction objectif.

Définition 3 (Modèle de recherche opérationnelle). Maximiser ou minimiser (*fonction objectif*) Sujet à $\{ \text{contraintes} \}$

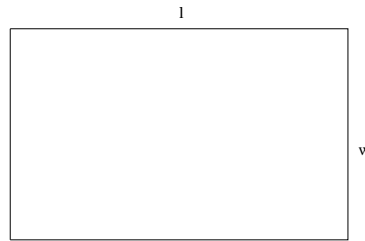
Variables : continues (réelles), entières, booléennes (0/1), ...

Objectif : linéaire / non-linéaire, concave / convexe, ...

Contraintes : linéaire / non-linéaire, concave / convexe, égalités / inégalités, ...

Paramètres : connus avec certitude (modèles déterministes) / incertains (modèles stochastiques)

Exemple 2 (Maximisation de la surface d'un rectangle). Supposons que l'on veut plier un fil de fer de longueur L en rectangle de manière à maximiser la surface du rectangle.



Formulation

$$\begin{array}{ll}\max & A = lw \\ \text{s.t.} & l + w = \frac{L}{2}\end{array}$$

Solution

- $A = \left(\frac{L}{2} - w\right) w = \frac{Lw}{2} - w^2$
- $\frac{dA}{dw} = \frac{L}{2} - 2w = 0$
- Solution optimale : $w = l = \frac{L}{4}$

Méthodes de résolution

- Dans l'exemple, *solution analytique* au problème.
- La plupart des problèmes pratiques sont trop grands ou trop complexes pour être résolus analytiquement.

Méthodes itératives

Déplacement de solution en solution pour atteindre l'optimum (*méthodes exactes*) ou une "bonne" solution (*heuristiques*).

- Importance des algorithmes et des solutions informatiques.

2 Quelques exemples de modèles mathématiques

Un problème de placement

Une entreprise veut placer un capital K . Six modalités de placement sont offertes sur le marché, les taux d'intérêt étant respectivement 3%, 2,5%, 3,5%, 4%, 5% et 4,5%.

Pour satisfaire aux contraintes légales, l'entreprise doit placer au moins 40% de son capital suivant les deux premières modalités, au plus 35% suivant les deux suivantes et au plus 35% suivant les deux dernières.

Combien l'entreprise doit-elle placer dans chaque modalité de façon à maximiser son intérêt total ?

Fabrication de crème glacée

Un fabricant désire produire 100 kg d'une préparation de base pour crème glacée. Cette préparation doit contenir 21,5 kg de matières grasses, 21 kg de sucre, 1,2 kg d'oeufs et 56,3 kg d'eau. Les ingrédients dont il dispose figurent ci-dessous, avec les pourcentages (en poids) de chaque constituant, et le coût au kg.

	Matières gr.	Sucre	Oeufs	Eau	Coût
Crème	40			60	3
Jaune d'oeuf frais	50		40	10	4
Lait	12			88	1
Jaune d'oeuf surgelé	30	14	40	16	2
Sirop		70		30	0,8
Eau				100	0

Le fabricant désire déterminer la composition du mélange de coût minimal.

Localisation d'émetteurs de télévision

Cinq sites ont été retenus pour construire des émetteurs de télévision destinés à desservir 10 localités. Le tableau ci-dessous donne, pour chaque site, le coût de construction d'un émetteur sur ce site et les localités desservies par cet émetteur.

Site	Coût	Localités desservies
1	4	1, 3, 5, 7, 8, 10
2	7	2, 4
3	11	1, 6, 8, 9, 10
4	6	1, 2, 4, 8, 10
5	10	3, 5, 6, 7, 9

Un problème de localisation

Une société de distribution désire construire un ou plusieurs entrepôts à partir desquels elle servira 6 clients importants. Quatre terrains appartenant à la société sont disponibles. Le tableau ci-dessous reprend, pour chacun d'eux, la capacité maximum de l'entrepôt qui pourrait y être construit, le coût de construction et le coût unitaire de transport de cet éventuel entrepôt vers chaque client. Sachant que les quantités demandées par les clients sont respectivement de 200, 250, 300, 100, 150, et 175, où faut-il construire un (des) entrepôt(s) et comment faut-il organiser la distribution de façon à minimiser le coût total de l'opération ?

Entrepôts	Clients						Capacité maximum	Coût construction
	1	2	3	4	5	6		
1	2	3	1	4	5	2	500	100
2	3	1	1	2	2	1	400	120
3	4	1	2	3	2	5	600	150
4	6	1	1	1	2	0	1000	80

3 Tour d'horizon des techniques de recherche opérationnelle

Recherche opérationnelle

La recherche opérationnelle est une technique d'*aide à la décision*.

Étapes pratiques

1. Définition du problème
2. Construction d'un modèle
3. Solution du modèle
4. Validation du modèle
5. Implémentation de la solution

Méthodologie

- Les étapes les plus importantes sont la définition du problème (suppose un *dialogue* avec le décideur) et la construction du modèle (prendre conscience des *hypothèses simplificatrices* et de leur impact).
- La phase de validation doit permettre de *remettre en cause la validité du modèle*.
- Une approche globale nécessite donc un aller-retour constant entre le modèle et les attentes du décideur.

Techniques principales

- Programmation linéaire
- Programmation en nombres entiers
- Optimisation dans les réseaux
- Programmation non linéaire
- "Optimisation" multi-critères
- Programmation dynamique
- Modèles stochastiques
- Simulation

Deuxième partie

Programmation linéaire

4 Définition, exemples et méthode de résolution

4.1 Notions de bases

Programmation linéaire

Définition 4 (Programme linéaire). Modèle mathématique dans lequel la fonction objectif et les contraintes sont linéaires en les variables.

Applications

Optimisation de l'usage de ressources limitées dans les domaines militaire, industriel, agricole, économique, ...

Existence d'algorithmes très efficaces pour résoudre des problèmes de très grande taille (*simplexe*, points intérieurs)

4.2 Exemples de modèles linéaires

Exemple 3 (Production de peinture). Une société produit de la peinture d'intérieur et d'extérieur à partir de deux produits de base M1 et M2.

Données

	Quantité utilisée par tonne		Quantité disponible par jour
	Extérieure	Intérieure	
M1	6	4	24
M2	1	2	6
Profit par tonne	5	4	

Contraintes supplémentaires

- Demande maximum en peinture d'intérieur : 2 tonnes / jour.
- La production en peinture d'intérieur ne dépasser que d'une tonne celle d'extérieur.

Formulation (Production de peinture)

Alternatives (*variables*, inconnues du problème)

x_1 = tonnes de peinture d'extérieur
produites par jour

x_2 = tonnes de peinture
d'intérieur produites par jour

Fonction objectif à optimiser

$$\max z = 5x_1 + 4x_2$$

Restrictions (*contraintes*)

$$\begin{aligned}
6x_1 + 4x_2 &\leq 24 \\
x_1 + 2x_2 &\leq 6 \\
x_2 &\leq 2 \\
x_2 - x_1 &\leq 1 \\
x_1, x_2 &\geq 0
\end{aligned}$$

Solutions et méthodes de résolution

— *Solution admissible* : satisfait toutes les contraintes.

$$x_1 = 3, x_2 = 1 (\Rightarrow z = 19)$$

— Nous voulons trouver la *solution (admissible) optimale*.
 — Infinité de solutions admissibles !

Méthodes pour trouver l'optimum

— Méthode graphique
 — Simplexe
 — (Ellipsoïde, points intérieurs)

Exemple 4 (Diet problem). — On désire déterminer la composition, à coût minimal, d'un aliment pour bétail qui est obtenu en mélangeant au plus trois produits bruts : orge et arachide.

— La quantité nécessaire par portion est de 400g.
 — L'aliment ainsi fabriqué devra comporter au moins 30% de protéines et au plus 5% de fibres.

Données

Aliment	Quantité par gramme d'aliment		Coût (EUR / kg)
	Protéines	Fibres	
Orge	0.09	0.02	1.5
Arachide	0.60	0.06	4.5

Formulation (Diet problem)

Variables

x_1 = grammes d'orge par portion
 x_2 = grammes d'arachide par portion

Objectif

$$\min z = 0.0015x_1 + 0.0045x_2$$

Contraintes

Quantité totale : $x_1 + x_2 \geq 400$

Protéines : $0.09x_1 + 0.6x_2 \geq 0.3(x_1 + x_2)$

Fibres : $0.02x_1 + 0.06x_2 \leq 0.05(x_1 + x_2)$

Non-négativité : $x_1, x_2 \geq 0$

4.3 Forme standard et forme canonique d'un programme linéaire

Forme standard

Définition 5 (Forme standard). Un programme linéaire est sous *forme standard* lorsque toutes ses contraintes sont des *égalités* et toutes ses variables sont non-négatives.

Représentation matricielle

$$\begin{array}{ll}\max & c^T x \\ \text{s.c.} & Ax = b \\ & x \geq 0\end{array}$$

n variables, m contraintes, $m < n$, $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$.

Forme canonique

Définition 6 (Forme canonique). Un programme linéaire est sous *forme canonique* lorsque toutes ses contraintes sont des *inégalités* et toutes ses variables sont non-négatives.

Représentation matricielle

$$\begin{array}{ll}\max & c^T x \\ \text{s.t.} & Ax \leq b \\ & x \geq 0\end{array}$$

n variables, m contraintes, $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$.

Théorème 1 (Equivalence des formes standard et canonique). Tout programme linéaire peut s'écrire sous forme standard et sous forme canonique.

Démonstration.

- Une contrainte d'inégalité $a^T x \leq b$ peut être transformée en égalité par l'introduction d'une *variable d'écart* :

$$\begin{array}{l}a^T x + s = b, \\ s \geq 0.\end{array}$$

- Une contrainte d'égalité $a^T x = b$ peut être remplacée par deux inégalités :

$$\begin{array}{l}a^T x \leq b \\ -a^T x \leq -b\end{array}$$

- $a^T x \geq b \Leftrightarrow -a^T x \leq -b$.
- $\min c^T x = -\max -c^T x$.
- Variable x non restreinte : substitution par deux variables (partie positive et négative)

$$\begin{array}{l}x = x^+ - x^- \\ x^+, x^- \geq 0.\end{array}$$

Il existe toujours une solution optimale telle que $x^+ = 0$ ou $x^- = 0$.

□

Forme standard du problème de production de peinture

$$\begin{aligned}\max z &= 5x_1 + 4x_2 \\ \text{s.c. } 6x_1 + 4x_2 &\leq 24 \\ x_1 + 2x_2 &\leq 6 \\ x_2 &\leq 2 \\ x_2 - x_1 &\leq 1 \\ x_1, x_2 &\geq 0\end{aligned}$$

Forme standard

$$\begin{aligned}\max z &= 5x_1 + 4x_2 \\ \text{s.c. } 6x_1 + 4x_2 + s_1 &= 24 \\ x_1 + 2x_2 + s_2 &= 6 \\ x_2 + s_3 &= 2 \\ -x_1 + x_2 + s_4 &= 1 \\ x_1, x_2, s_1, s_2, s_3, s_4 &\geq 0\end{aligned}$$

Forme matricielle

$$\begin{aligned}\max \quad & c^T x \\ \text{s.t. } & Ax = b \\ & x \geq 0\end{aligned}$$

$$c = \begin{pmatrix} 5 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}, A = \begin{pmatrix} 6 & 4 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 24 \\ 6 \\ 2 \\ 1 \end{pmatrix}$$

Variables pouvant prendre des valeurs négatives

Exemple 5 (Vente de hamburgers).

- Un fast-food vend des hamburgers et des cheeseburgers. Un hamburger utilise 125 g. de viande alors qu'un cheeseburger n'en utilise que 100 g.
- Le fast-food démarre chaque journée avec 10 kg de viande mais peut commander de la viande supplémentaire avec un coût additionnel de 2 EUR par kg pour la livraison.
- Le profit est de 0.02 EUR pour un hamburger et 0.015 EUR pour un cheeseburger.
- La demande ne dépasse pas 900 sandwiches / jour, et les surplus de viande sont donnés au Restos du Coeur.

Combien le fast-food doit-il produire de sandwiches de chaque type par jour ?

Variables

x_1 = nombre de hamburgers / jour x_2 = nombre de cheeseburgers / jour

Contraintes

- Commande de viande supplémentaire :

$$125x_1 + 100x_2 + x_3 = 10000, \quad x_3 \text{ non restreint}$$

- Le coût pour la viande supplémentaire apparaît seulement si $x_3 < 0$.

— Substitution de x_3 par 2 variables non-négatives :

$$x_3 = x_3^+ - x_3^-, \quad x_3^+, x_3^- \geq 0$$

$$125x_1 + 100x_2 + x_3^+ - x_3^- = 10000$$

— Borne supérieure sur les ventes : $x_1 + x_2 \leq 900$.

Modèle complet

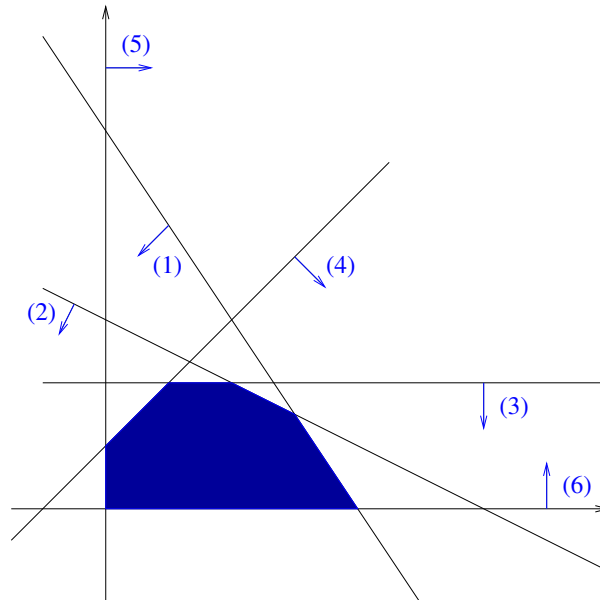
$$\begin{aligned} \max z = & 0.02x_1 + 0.015x_2 - 0.002x_3^- \\ \text{s.c.} \quad & 125x_1 + 100x_2 + x_3^+ - x_3^- = 10000 \\ & x_1 + x_2 \leq 900 \\ & x_1, x_2, x_3^+, x_3^- \geq 0 \end{aligned}$$

Remarque : Il existe une solution optimale telle que $x_3^+ = 0$ ou $x_3^- = 0$.

4.4 Résolution de programmes linéaires

4.4.1 Résolution graphique

Représentation graphique



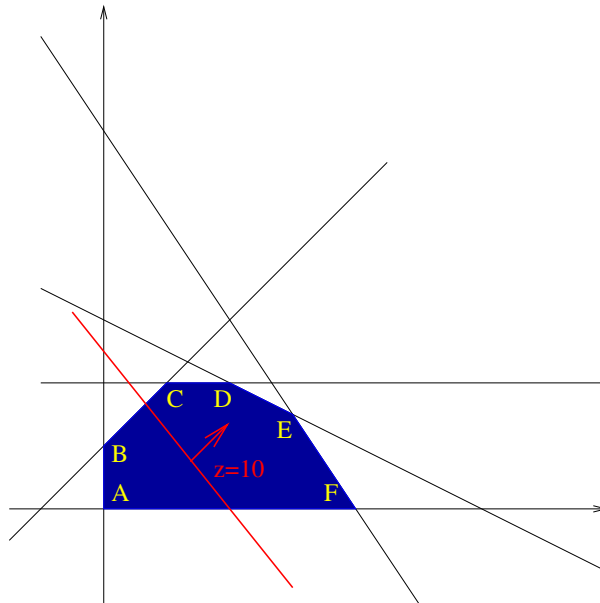
Production de peinture

$$\max z = 5x_1 + 4x_2$$

sous les contraintes :

$$\begin{aligned} 6x_1 + 4x_2 &\leq 24 & (1) \\ x_1 + 2x_2 &\leq 6 & (2) \\ x_2 &\leq 2 & (3) \\ x_2 - x_1 &\leq 1 & (4) \\ x_1 &\geq 0 & (5) \\ x_2 &\geq 0 & (6) \end{aligned}$$

Géométrie des solutions



Ensemble des solutions admissibles

Polyèdre (ABCDEF)

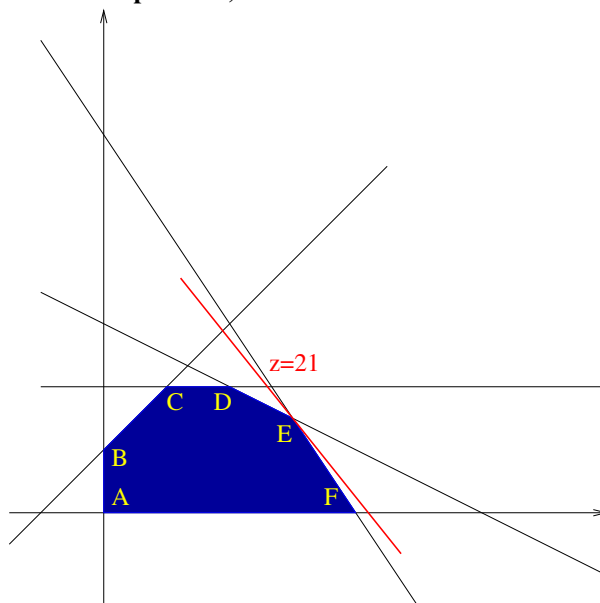
Courbes de niveaux de l'objectif

Ensemble de solutions ayant un profit (valeur de l'objectif) donné : intersection entre une droite et le polyèdre.

Amélioration de la solution

Recherche d'une direction dans laquelle le profit z augmente.

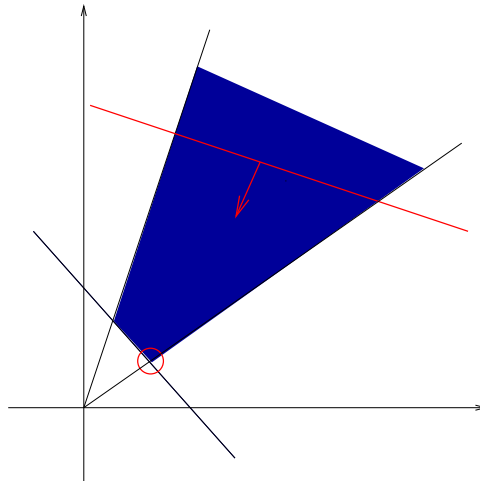
Résolution graphique (Production de peinture)



Recherche de la solution optimale

- La droite mobile doit garder une intersection avec l'ensemble des solutions admissibles.
- Solution optimale : $x_1 = 3$, $x_2 = 1.5$ (E)
- La solution optimale est un *sommet du polyèdre*.
- Cette observation est la base de l'algorithme du simplexe.

Résolution graphique (Diet problem)



Diet problem

$$\min z = 0.0015x_1 + 0.0045x_2$$

sous les contraintes

$$\begin{aligned}x_1 + x_2 &\geq 400 \\0.21x_1 - 0.30x_2 &\leq 0 \\0.03x_1 - 0.01x_2 &\geq 0 \\x_1 &\geq 0 \\x_2 &\geq 0\end{aligned}$$

Solution optimale

$$x_1 = \frac{4000}{17} \simeq 235.3 \quad x_2 = \frac{2800}{17} \simeq 164.7 \quad z = \frac{186}{170} \simeq 1.094$$

4.4.2 La méthode du simplexe

Idées de base

- Solution optimale : *sommet (point extrême)*.
- Idée fondamentale du simplexe : déplacement de sommet en sommet adjacent de manière à améliorer la fonction objectif.
- Transformation des inégalités en égalités : *forme standard* du programme linéaire - système de m équations à n inconnues ($m < n$).
- Identification algébrique des sommets : correspondance avec les bases d'un système d'équations.

Solutions de base

- Système de m équations linéaires à n inconnues ($m < n$) : infinité de solutions.
- Si on fixe à zéro $n - m$ variables : système de m équations à m inconnues possédant une solution unique (si la matrice est inversible). C'est une *solution de base*.

Définition 7 (Solution de base). Une *solution de base* d'un programme linéaire est la solution unique du système de m équations à m inconnues obtenu en fixant à zéro $n - m$ variables (pourvu que la matrice du système soit inversible).

Les variables fixées à zéro sont appelées *variables hors base* et les autres *variables en base*.

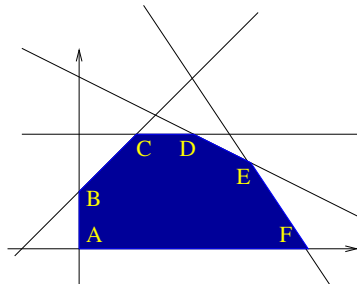
Exemple 6 (Production de peinture). Prenons $B = \{s_1, s_2, s_3, s_4\}$.

$$\begin{aligned} z &= 0 + 5x_1 + 4x_2 \\ s_1 &= 24 - 6x_1 - 4x_2 \\ s_2 &= 6 - x_1 - 2x_2 \\ s_3 &= 2 - x_2 \\ s_4 &= 1 + x_1 - x_2 \end{aligned}$$

Si $x_1 = x_2 = 0$, alors $s_1 = 24$, $s_2 = 6$, $s_3 = 2$, $s_4 = 1$. Toutes ces valeurs sont non-négatives et la solution est réalisable.

Définition 8 (Solution de base réalisable). Une solution de base telle que toutes les variables prennent des valeurs non-négatives est appelée *solution de base réalisable*.

Géométrie des solutions de base



- Prenons $B = \{s_1, s_2, s_3, s_4\} \Rightarrow x_1 = x_2 = 0$, $s_1 = 24$, $s_2 = 6$, $s_3 = 2$, $s_4 = 1$.
- Cette solution de base réalisable correspond au sommet $(0, 0)$.

Base	Solution	Objectif	Sommet
$\{s_1, s_2, s_3, s_4\}$	$(0, 0)$	0	A
$\{x_1, s_2, s_3, s_4\}$	$(4, 0)$	20	F
$\{s_1, x_1, s_3, s_4\}$	$(6, 0)$	—	Non réalisable
$\{x_1, x_2, s_3, s_4\}$	$(3, 1.5)$	21	E

Théorème 2. Toute solution de base réalisable correspond à un sommet du polyèdre.

Détermination de la solution de base optimale

- Nombre maximum de solutions de base : $\frac{n!}{m!(n-m)!}$
- Algorithme "bête et méchant" : énumération de toutes les bases.
- Méthode du simplexe : partir d'une solution de base admissible et passer à une solution de base voisine qui améliore la valeur de l'objectif.
- Solution voisine : changement d'une variable en base.
- 3 étapes :
 1. Détermination de la variable entrante.
 2. Détermination de la variable sortante.
 3. Pivotage.

L'algorithme du simplexe

Variable entrante

$$\begin{aligned} z &= 0 + 5x_1 + 4x_2 \\ s_1 &= 24 - 6x_1 - 4x_2 \\ s_2 &= 6 - x_1 - 2x_2 \\ s_3 &= 2 - x_2 \\ s_4 &= 1 + x_1 - x_2 \end{aligned}$$

- Si x_1 (ou x_2) augmente (*entre en base*), la valeur de la fonction objectif z augmente.
- Quelle est la valeur maximale de x_1 ?

— Contraintes : les autres variables doivent rester positives.

Variable sortante

$$\begin{aligned} s_1 &= 24 - 6x_1 \geq 0 \rightarrow x_1 \leq 4 \\ s_2 &= 6 - x_1 \geq 0 \rightarrow x_1 \leq 6 \\ s_3 &= 2 \geq 0 \rightarrow 2 \geq 0 \quad \text{toujours!} \\ s_4 &= 1 + x_1 \geq 0 \rightarrow x_1 \geq -1 \quad \text{toujours!} \end{aligned} \quad \Rightarrow x_1 \leq 4$$

Pivotage

- Si $x_1 = 4$, alors $s_1 = 0$.
- x_1 entre en base, s_1 sort de la base.
- Substitution :

$$x_1 = 4 - \frac{1}{6}s_1 - \frac{2}{3}x_2$$

— Nouveau système :

$$\begin{aligned} z &= 20 - \frac{5}{6}s_1 + \frac{2}{3}x_2 \\ x_1 &= 4 - \frac{1}{6}s_1 - \frac{2}{3}x_2 \\ s_2 &= 2 + \frac{1}{6}s_1 - \frac{4}{3}x_2 \\ s_3 &= 2 - x_2 \\ s_4 &= 5 - \frac{1}{6}s_1 - \frac{5}{3}x_2 \end{aligned}$$

Représentation matricielle (dictionnaire)

- B = indices des variables en base, N = indices des variables hors base.
- “Décomposition” des données et variables (en réarrangeant l’ordre des variables) :

$$A = \begin{pmatrix} A_B & A_N \end{pmatrix}, x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$$

avec $x_B, c_B \in \mathbb{R}^m$, $x_N, c_N \in \mathbb{R}^{n-m}$, $A_B \in \mathbb{R}^{m \times m}$, $A_N \in \mathbb{R}^{m \times (n-m)}$.

Equations du simplexe

Si A_B est inversible :

$$\begin{aligned} Ax = b &\Leftrightarrow A_B x_B + A_N x_N = b \\ &\Leftrightarrow x_B + A_B^{-1} A_N x_N = A_B^{-1} b \end{aligned}$$

$$\begin{aligned} z = c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T (A_B^{-1} b - A_B^{-1} A_N x_N) + c_N^T x_N \\ &= c_B^T A_B^{-1} b + \underbrace{(c_N^T - c_B^T A_B^{-1} A_N)}_{\text{coûts réduits ou profits marginaux}} x_N \end{aligned}$$

Notation :

$$\begin{aligned} x_l + \sum_{k \in N} \bar{a}_{lk} x_k &= \bar{b}_l \quad l \in B \\ z &= \bar{z} + \sum_{k \in N} \bar{c}_k x_k \end{aligned}$$

Règles de pivotage

Variable entrante

Choisir la variable k hors base avec le profit marginal maximum ($\max z$) ou le coût réduit minimum ($\min z$).

$$\begin{aligned}\max z &\rightarrow k = \arg \max_{i \in N} \bar{c}_i \\ \min z &\rightarrow k = \arg \min_{i \in N} \bar{c}_i\end{aligned}$$

Si $\bar{c}_k \leq 0$ (max) ou $\bar{c}_k \geq 0$ (min) pour tout $k \in N$, *solution optimale*, STOP.

$$\begin{aligned}z &= 0 \quad \boxed{+5x_1} + 4x_2 \\ s_1 &= 24 \quad -6x_1 \quad -4x_2 \\ s_2 &= 6 \quad -x_1 \quad -2x_2 \\ s_3 &= 2 \quad \quad -x_2 \\ s_4 &= 1 \quad +x_1 \quad -x_2\end{aligned}$$

Variable sortante

Choisir la variable l en base telle que

$$l = \arg \min_{j \in B: \bar{a}_{jk} > 0} \frac{\bar{b}_j}{\bar{a}_{jk}}$$

Si $\bar{a}_{lk} \leq 0$ pour tout $l \in B$, *problème non borné*, STOP.

$$\begin{aligned}z &= 0 \quad +5x_1 + 4x_2 \\ s_1 &= 24 \quad \boxed{-6x_1} - 4x_2 \\ s_2 &= 6 \quad -x_1 \quad -2x_2 \\ s_3 &= 2 \quad \quad -x_2 \\ s_4 &= 1 \quad +x_1 \quad -x_2\end{aligned}$$

Pivotage

$$\begin{aligned}\bar{a}_{ij}' &= \begin{cases} \frac{\bar{a}_{lj}}{\bar{a}_{lk}} & i = l \\ \bar{a}_{ij} - \frac{\bar{a}_{ik}\bar{a}_{lj}}{\bar{a}_{lk}} & i \neq l \end{cases} \\ \bar{b}_i' &= \begin{cases} \frac{\bar{b}_l}{\bar{a}_{lk}} & i = l \\ \bar{b}_i - \frac{\bar{a}_{ik}\bar{b}_l}{\bar{a}_{lk}} & i \neq l \end{cases}\end{aligned}$$

$$\begin{aligned}z &= 0 \quad +5x_1 + 4x_2 \\ s_1 &= 24 \quad \boxed{-6x_1} - 4x_2 \\ s_2 &= 6 \quad -x_1 \quad -2x_2 \\ s_3 &= 2 \quad \quad -x_2 \\ s_4 &= 1 \quad +x_1 \quad -x_2\end{aligned}$$

$$\begin{aligned}z &= 20 - \frac{5}{6}s_1 + \frac{2}{3}x_2 \\ x_1 &= 4 - \frac{1}{6}s_1 - \frac{2}{3}x_2 \\ s_2 &= 2 + \frac{1}{6}s_1 \quad \boxed{-\frac{4}{3}x_2} \\ s_3 &= 2 \quad \quad -x_2 \\ s_4 &= 5 - \frac{1}{6}s_1 - \frac{5}{3}x_2\end{aligned}$$

$$\begin{aligned}z &= 21 - \frac{3}{4}s_1 - \frac{1}{2}s_2 \\ x_1 &= 3 - \frac{1}{4}s_1 + \frac{1}{2}s_2 \\ x_2 &= \frac{3}{2} + \frac{1}{8}s_1 - \frac{3}{4}s_2 \\ s_3 &= \frac{1}{2} - \frac{1}{8}s_1 + \frac{3}{4}s_2 \\ s_4 &= \frac{5}{2} - \frac{3}{8}s_1 + \frac{5}{4}s_2\end{aligned}$$

Présentation en tableau

Présentation compacte pour effectuer les calculs sans répéter les systèmes d'équations.

Var. en base	z	x	Solution
z	-1	$c^T - c^T A_B^{-1} A = \bar{c}$	$-c_B^T A_B^{-1} b$
x_B	0	$A_B^{-1} A = \bar{a}$	$A_B^{-1} b$

Itération 1

Var. en base	z	x_1	x_2	s_1	s_2	s_3	s_4	Solution
z	-1	5	4	0	0	0	0	0
s_1	0	6	4	1	0	0	0	24
s_2	0	1	2	0	1	0	0	6
s_3	0	0	1	0	0	1	0	2
s_4	0	-1	1	0	0	0	1	1

Itération 2

Var. en base	z	x_1	x_2	s_1	s_2	s_3	s_4	Solution
z	-1	0	$\frac{2}{3}$	$-\frac{5}{6}$	0	0	0	-20
x_1	0	1	$\frac{2}{3}$	$\frac{1}{6}$	0	0	0	4
s_2	0	0	$\frac{4}{3}$	$-\frac{1}{6}$	1	0	0	2
s_3	0	0	1	0	0	1	0	2
s_4	0	0	$\frac{5}{3}$	$\frac{1}{6}$	0	0	1	5

Itération 3

Var. en base	z	x_1	x_2	s_1	s_2	s_3	s_4	Solution
z	-1	0	0	$-\frac{3}{4}$	$-\frac{1}{2}$	0	0	-21
x_1	0	1	0	$\frac{1}{4}$	$-\frac{1}{2}$	0	0	3
x_2	0	0	1	$-\frac{1}{8}$	$\frac{3}{4}$	0	0	$\frac{3}{2}$
s_3	0	0	0	$\frac{1}{8}$	$-\frac{3}{4}$	1	0	$\frac{1}{2}$
s_4	0	0	0	$\frac{3}{8}$	$-\frac{5}{4}$	0	1	$\frac{5}{2}$

4.4.3 La méthode des deux phases

Solution initiale artificielle

- Une solution de base admissible n'est pas toujours connue a priori.
- Certains problèmes n'admettent pas de solution admissible, donc il est impossible de trouver une base de départ.
- La méthode des *deux phases* va permettre de déterminer une base admissible ou prouver que le problème est impossible.

Exemple 7 (Méthode des 2 phases).

$$\begin{array}{ll}
 \min & z = 4x_1 + x_2 \\
 \text{s.c.} & 3x_1 + x_2 = 3 \\
 & 4x_1 + 3x_2 \geq 6 \\
 & x_1 + 2x_2 \leq 4 \\
 & x_1, x_2 \geq 0
 \end{array}$$

Introduction des variables d'écart

$$\begin{array}{llllll}
 \min & z = & 4x_1 & +x_2 & & \\
 \text{s.c.} & & 3x_1 & +x_2 & & = 3 \\
 & & 4x_1 & +3x_2 & -x_3 & = 6 \\
 & & x_1 & +2x_2 & & +x_4 = 4 \\
 & & x_1, & x_2, & x_3, & x_4 \geq 0
 \end{array}$$

- Pas de base admissible "triviale".
- On voudrait voir apparaître une *matrice identité*.
- Introduction de *variables artificielles*.

Introduction des variables artificielles

$$\begin{array}{llllllll}
 \min & z = & 4x_1 & +x_2 & & & & \\
 \min & r = & & & & R_1 & +R_2 & \\
 \min & r = & -7x_1 & -4x_2 & +x_3 & & & +9 \\
 \text{s.c.} & & 3x_1 & +x_2 & & +R_1 & & = 3 \\
 & & 4x_1 & +3x_2 & -x_3 & & +R_2 & = 6 \\
 & & x_1 & +2x_2 & & & & +x_4 = 4 \\
 & & x_1, & x_2, & x_3, & R_1, & R_2, & x_4 \geq 0
 \end{array}$$

- R_1, R_2 et x_4 peuvent être utilisées comme base de départ admissible.
- Base pour le système de départ si $R_1 = R_2 = 0$ (hors base).
- Réécrire l'objectif en fonction des variables hors base !

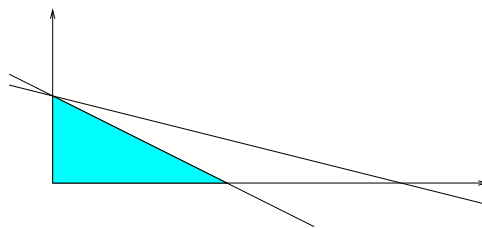
4.4.4 Cas particuliers

Bases dégénérées

- Si plusieurs choix sont possibles pour la variable sortante, une des variables en base sera nulle à l'itération suivante.
- Base telle qu'une variable en base est nulle : *base dégénérée*.
- Correspond à une *contrainte redondante*.
- Risque de cyclage !

Exemple 8 (Bases dégénérées).

$$\begin{array}{llll}
 \max & z = & 3x_1 & +9x_2 \\
 \text{s.c.} & & x_1 & +4x_2 \leq 8 \\
 & & x_1 & +2x_2 \leq 4 \\
 & & x_1, & x_2 \geq 0
 \end{array}$$



Var. en base	z	x_1	x_2	s_1	s_2	Solution
z	-1	3	9	0	0	0
s_1	0	1	4	1	0	8
s_2	0	1	2	0	1	4
z	-1	$\frac{3}{4}$	0	$-\frac{9}{4}$	0	-18
x_2	0	$\frac{1}{4}$	1	$\frac{1}{4}$	0	2
s_2	0	$\frac{1}{2}$	0	$-\frac{1}{2}$	1	0
z	-1	0	0	$-\frac{3}{2}$	$-\frac{3}{2}$	-18
x_2	0	0	1	$\frac{1}{2}$	$-\frac{1}{2}$	2
x_1	0	1	0	-1	2	0

- En pratique, le risque de cyclage est quasiment nul.
- Méthodes pour l'éviter : perturbations, ordre lexicographique pour les variables entrant en base, ...
- Une bonne analyse a priori pour éliminer les redondances est le meilleur moyen d'éviter ces problèmes !

Solutions optimales multiples

- Si la fonction objectif est parallèle à une contrainte active pour la solution optimale, la même valeur de l'objectif peut être prise par plusieurs solutions admissibles.
- Il y a une *infinité* de solutions optimales dans ce cas (toutes les combinaisons convexes de sommets optimaux).
- Cela se traduit par un *profit marginal nul* pour une ou plusieurs variables hors base.

Exemple 9 (Solutions optimales multiples).

$$\begin{array}{ll}
 \max & z = 2x_1 + 4x_2 \\
 \text{s.c.} & x_1 + 2x_2 \leq 5 \\
 & x_1 + x_2 \leq 4 \\
 & x_1, x_2 \geq 0
 \end{array}$$

Var. en base	z	x_1	x_2	s_1	s_2	Solution
z	-1	2	4	0	0	0
s_1	0	1	2	1	0	5
s_2	0	1	1	0	1	4
z	-1	0	0	-2	0	-10
x_2	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0	$\frac{5}{2}$
s_2	0	$\frac{1}{2}$	0	$-\frac{1}{2}$	1	$\frac{3}{2}$
z	-1	0	0	-2	0	-10
x_2	0	0	1	1	-1	1
x_1	0	1	0	-1	2	3

Solution optimale :

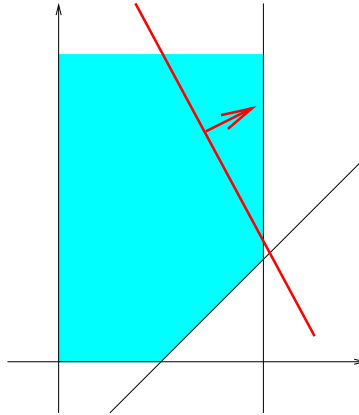
$$\begin{aligned}
 x_1 &= 0\alpha + 3(1 - \alpha) = 3 - 3\alpha \\
 x_2 &= \frac{5}{2}\alpha + 1(1 - \alpha) = 1 + \frac{3}{2}\alpha \quad (0 \leq \alpha \leq 1)
 \end{aligned}$$

Problèmes non bornés

- Certains problèmes sont *non bornés* dans une direction donnée.
- Si cette direction est une direction d'amélioration de la fonction objectif, celle-ci peut prendre une valeur arbitrairement grande !

Exemple 10 (Problèmes non bornés).

$$\begin{array}{llllll} \max & z = & 2x_1 & +x_2 & & \\ \text{s.c.} & & x_1 & -x_2 & \leq & 1 \\ & & 2x_1 & & \leq & 4 \\ & & x_1, & x_2 & \geq & 0 \end{array}$$



Var. en base	z	x_1	x_2	s_1	s_2	Solution
z	-1	2	1	0	0	0
s_1	0	1	-1	1	0	1
s_2	0	2	0	0	1	4

- Tous les coefficients (sauf le profit marginal) dans la colonne de x_2 sont négatifs ou nuls.
- Cela signifie que toutes les contraintes de non-négativité sont satisfaites quelle que soit la valeur de x_2 .
- L'objectif peut donc augmenter indéfiniment.

Problèmes impossibles

- Le système de contraintes peut n'avoir aucune solution.
- Méthode des 2 phases : termine avec des variables artificielles non nulles.
- Généralement, provient d'une mauvaise formulation du problème.

Exemple 11 (Problèmes impossibles).

$$\begin{array}{llllll} \max & z = & 3x_1 & +2x_2 & & \\ \text{s.c.} & & 2x_1 & +x_2 & \leq & 2 \\ & & 3x_1 & +4x_2 & \geq & 12 \\ & & x_1, & x_2 & \geq & 0 \end{array}$$

5 Dualité

5.1 Le problème dual

Problème primal et problème dual

Problème primal

$$\begin{array}{ll}\max & c^T x \\ \text{s.c.} & Ax = b \\ & x \geq 0\end{array}$$

n variables, m contraintes, $m < n$, $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$.

Problème dual

$$\begin{array}{ll}\min & b^T y \\ \text{s.c.} & A^T y \geq c \\ & (y \text{ non restreint})\end{array}$$

m variables, n contraintes, $m < n$, $c \in \mathbb{R}^n$, $b, y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$.

Exemple 12 (Problème primal et dual - forme standard).

Problème primal :

$$\begin{array}{llll}\max & z = & x_1 & +x_2 \\ \text{s.c.} & & 2x_1 & +x_2 = 5 \quad (y_1) \\ & & 3x_1 & -x_2 = 6 \quad (y_2) \\ & & x_1, & x_2 \geq 0\end{array}$$

Problème dual :

$$\begin{array}{llll}\min & w = & 5y_1 & +6y_2 \\ \text{s.c} & & 2y_1 & +3y_2 \geq 1 \quad (x_1) \\ & & y_1 & -y_2 \geq 1 \quad (x_2)\end{array}$$

Propriétés et règles de construction du dual

Théorème 3. Le problème dual du problème dual est le problème primal.

Règles de construction

Problème max	Problème min
Contrainte	Variable
\leq	≥ 0
$=$	non restreinte
Variable	Contrainte
≥ 0	\geq
non restreinte	$=$

Exemple 13 (Problème primal et dual - forme générale).

Problème primal :

$$\begin{array}{llll}\max & z = & 5x_1 & +12x_2 & +4x_3 \\ \text{s.c.} & & x_1 & +2x_2 & +x_3 \leq 10 \quad (y_1) \\ & & 2x_1 & -x_2 & +3x_3 = 8 \quad (y_2) \\ & & x_1, & x_2, & x_3 \geq 0\end{array}$$

Problème dual :

$$\begin{array}{llll} \min & w = & 10y_1 & +8y_2 \\ \text{s.c.} & & y_1 & +2y_2 \geq 5 \quad (x_1) \\ & & 2y_1 & -y_2 \geq 12 \quad (x_2) \\ & & y_1 & +3y_2 \geq 4 \quad (x_3) \\ & & y_1 & \geq 0 \end{array}$$

5.2 Relations primal/dual

Théorème 4 (Dualité faible). Considérons la paire primale-duale :

$$\begin{array}{ll} \max & c^T x \\ \text{s.c.} & Ax = b \\ & x \geq 0 \end{array}$$

$$\begin{array}{ll} \min & b^T y \\ \text{s.c.} & A^T y \geq c \end{array}$$

— Si x est une solution admissible du primal et y une solution admissible du dual, alors

$$c^T x \leq b^T y$$

— S'il y a égalité, alors x est une solution optimale du primal et y une solution optimale du dual.

Théorème 5 (Dualité forte). Considérons la paire primale-duale :

$$\begin{array}{ll} \max & c^T x \\ \text{s.c.} & Ax = b \\ & x \geq 0 \end{array}$$

$$\begin{array}{ll} \min & b^T y \\ \text{s.c.} & A^T y \geq c \end{array}$$

— Si le primal et le dual admettent tous les deux une solution admissible, ils ont tous deux une solution optimale finie et la même valeur objectif optimale.

— Si le primal (dual) est non borné, le dual (primal) n'admet pas de solution admissible.

Théorème 6 (Complémentarité). Considérons la paire primale-duale :

$$\begin{array}{ll} \max & c^T x \\ \text{s.c.} & Ax = b \\ & x \geq 0 \end{array}$$

$$\begin{array}{ll} \min & b^T y \\ \text{s.c.} & A^T y \geq c \end{array}$$

Si x est une solution optimale du primal et y une solution optimale du dual, alors

$$x_i(a_i^T y - c_i) = 0.$$

où a_i est la i -ème colonne de A .

En d'autres termes :

$$\begin{aligned} x_i > 0 &\Rightarrow a_i^T y = c_i, \\ a_i^T y > c_i &\Rightarrow x_i = 0. \end{aligned}$$

Exemple 14 (Résolution du dual par les règles de complémentarité).

Primal (P) :

$$\begin{array}{llll} \max & z = & 5x_1 & +12x_2 & +4x_3 \\ \text{s.c.} & & x_1 & +2x_2 & +x_3 \leq 10 \quad (y_1) \\ & & 2x_1 & -x_2 & +3x_3 = 8 \quad (y_2) \\ & & x_1, & x_2, & x_3 \geq 0 \end{array}$$

Dual (D) :

$$\begin{array}{llll} \min & w = & 10y_1 & +8y_2 \\ \text{s.c} & & y_1 & +2y_2 \geq 5 \quad (x_1) \\ & & 2y_1 & -y_2 \geq 12 \quad (x_2) \\ & & y_1 & +3y_2 \geq 4 \quad (x_3) \\ & & y_1 & \geq 0 \end{array}$$

Solution optimale de (P) :

$$\begin{aligned} (x_1, x_2, x_3) &= \left(\frac{26}{5}, \frac{12}{5}, 0 \right) \\ z &= \frac{274}{5} \end{aligned}$$

$$\begin{aligned} x_1 > 0 &\Rightarrow y_1 + 2y_2 = 5 \\ x_2 > 0 &\Rightarrow 2y_1 - y_2 = 12 \end{aligned}$$

Solution optimale de (D) :

$$\begin{aligned} (y_1, y_2) &= \left(\frac{29}{5}, -\frac{2}{5} \right) \\ w &= \frac{274}{5} \end{aligned}$$

5.3 Interprétation économique de la dualité

- La *forme canonique* d'un programme linéaire peut être interprétée comme un problème d'*allocation de ressources*.
- Paire primale-duale :

$$\begin{array}{ll} \max & c^T x \\ \text{s.c.} & Ax \leq b \\ & x \geq 0 \end{array}$$

$$\begin{array}{ll} \min & b^T y \\ \text{s.c.} & A^T y \geq c \\ & y \geq 0 \end{array}$$

- Données :
 - c_j : profit par unité d'activité j .
 - b_i : disponibilité de la ressource i .
 - a_{ij} : consommation de la ressource i par unité d'activité j .
- Variables :
 - x_j : niveau de l'activité j .
 - y_i : valeur d'une unité de la ressource i .

Interprétation de la dualité faible

$$z \leq w : \quad \text{profit} \leq \text{valeur des ressources}$$

Interprétation de la dualité forte

Le profit maximal est atteint si les ressources ont été exploitées complètement, i.e. jusqu'à épuisement de leur valeur.

Exemple 15 (Dualité dans le problème de production de peinture).

$$\begin{aligned}
 \max z &= 5x_1 + 4x_2 \\
 \text{s.c.} \quad &6x_1 + 4x_2 \leq 24 \\
 &x_1 + 2x_2 \leq 6 \\
 &x_2 \leq 2 \\
 &-x_1 + x_2 \leq 1 \\
 &x_1, x_2 \geq 0
 \end{aligned}$$

$$\begin{aligned}
 \min w &= 24y_1 + 6y_2 + 2y_3 + y_4 \\
 &6y_1 + y_2 - y_4 \geq 5 \\
 &4y_1 + 2y_2 + y_3 + y_4 \geq 4 \\
 &y_1, y_2, y_3, y_4 \geq 0
 \end{aligned}$$

$$x_1 = 3, x_2 = 1.5, z = 21$$

$$y_1 = 0.75, y_2 = 0.5, y_3 = y_4 = 0, w = 21$$

- Le profit augmente de 0.75 par augmentation d'une tonne de M1 et de 0.5 par tonne de M2.
- Les "ressources" 3 et 4 sont abondantes, augmenter ces ressources n'apporte aucun profit supplémentaire.

5.4 Méthode simplexe duale

Relations primal / dual dans le tableau du simplexe

Tableau primal final

Var. en base	z	x_1	x_2	s_1	s_2	s_3	s_4	Solution
z	-1	0	0	$-\frac{3}{4}$	$-\frac{1}{2}$	0	0	-21
x_1	0	1	0	$\frac{1}{4}$	$-\frac{1}{2}$	0	0	3
x_2	0	0	1	$-\frac{1}{8}$	$\frac{3}{4}$	0	0	$\frac{3}{2}$
s_3	0	0	0	$\frac{1}{8}$	$-\frac{3}{4}$	1	0	$\frac{1}{2}$
s_4	0	0	0	$\frac{3}{8}$	$-\frac{5}{4}$	0	1	$\frac{5}{2}$

Tableau dual final

Var. en base	w	t_1	t_2	y_1	y_2	y_3	y_4	Solution
w	-1	3	$\frac{3}{2}$	0	0	$\frac{1}{2}$	$\frac{5}{2}$	-21
y_1	0	$-\frac{1}{4}$	$\frac{1}{8}$	1	0	$-\frac{1}{8}$	$-\frac{3}{8}$	$\frac{3}{4}$
y_2	0	$\frac{1}{2}$	$-\frac{3}{4}$	0	1	$\frac{3}{4}$	$\frac{5}{4}$	$\frac{1}{2}$

- Ici, toutes les contraintes dans le primal et dans le dual sont des inégalités \Rightarrow correspondance entre variable originale du primal et variable d'écart du dual (et vice-versa).

Primal	x_1	x_2	s_1	s_2	s_3	s_4
Dual	t_1	t_2	y_1	y_2	y_3	y_4

- Observation :
 - Les valeurs des variables duales sont égales à l'opposé des profits marginaux des variables correspondantes du primal.
 - Les valeurs des variables primales sont égales aux coûts réduits des variables correspondantes du dual.
 - Conséquence de la *complémentarité* des solutions primale et duale.

Théorème 7. Considérons la paire primale-duale :

$$\begin{aligned} \max \quad & c^T x \\ \text{s.c.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & b^T y \\ \text{s.c.} \quad & A^T y \geq c \end{aligned}$$

1. Pour toute solution de base non dégénérée x du primal, il existe une et une seule solution y du dual complémentaire à x .
2. x est admissible si et seulement si y satisfait les conditions d'optimalité (coûts réduits non négatifs).
3. y est admissible si et seulement si x satisfait les conditions d'optimalité (profits marginaux négatifs ou nuls).

Démonstration. — Le dual peut se réécrire sous forme standard :

$$\begin{aligned} \min \quad & b^T y \\ \text{s.c.} \quad & A^T y - t = c \\ & t \geq 0 \end{aligned}$$

- Considérons un ensemble de variables en base B et un ensemble de variables hors base N , tels que la base correspondante est non dégénérée :

$$x_B = A_B^{-1}b > 0 \quad x_N = 0$$

- Par complémentarité, $t_B = 0$ et $A_B^T y = c_B$. Dès lors,

$$y^T = c_B^T A_B^{-1}, \quad t_B^T = 0 \quad t_N^T = -(c_N^T - c_B^T A_B^{-1} A_N)$$

- (2) et (3) sont des conséquences directes.

□

Simplexe dual : motivation

- Il arrive souvent qu'une solution en base *non admissible*, mais satisfaisant les contraintes d'optimalité soit identifiable facilement (par exemple, variables d'écart de contraintes \geq).
- Cette base correspond à une solution admissible du dual.
- Idée de la méthode simplexe duale : résoudre (implicitement) le dual par la méthode du simplexe (mais en travaillant sur le tableau primal !)
- Partir avec une solution de base satisfaisant les conditions d'optimalité (= base admissible pour le dual) et chercher à la rendre admissible (= dual optimale).

Règles de pivotage (simplexe dual)

Variable sortante

Choisir la variable l en base avec la valeur minimum (négative). Si $\bar{b}_l \geq 0$ pour tout $l \in B$, *solution admissible*, STOP.

Variable entrante

Choisir la variable k hors base telle que

$$k = \arg \min_{i \in N: \bar{a}_{li} < 0} - \frac{|\bar{c}_i|}{\bar{a}_{li}}$$

Si $\bar{a}_{li} \geq 0$ pour tout $i \in N$, *problème non admissible*, STOP.

Exemple 16 (Simplexe dual).

$$\begin{array}{llll} \min z = & 3x_1 & +2x_2 & \\ \text{s.c.} & 3x_1 & +x_2 & \geq 3 \\ & 4x_1 & +3x_2 & \geq 6 \\ & x_1 & +x_2 & \leq 3 \\ & x_1, & x_2 & \geq 0 \end{array}$$

Forme standard avec les variables d'écart en base :

$$\begin{array}{llllllll} \min z = & 3x_1 & +2x_2 & & & & & \\ \text{s.c.} & -3x_1 & -x_2 & +s_1 & & & & = -3 \\ & -4x_1 & -3x_2 & & +s_2 & & & = -6 \\ & x_1 & +x_2 & & & +s_3 & & = 3 \\ & x_1, & x_2, & s_1, & s_2, & s_3 & & \geq 0 \end{array}$$

Var. en base	z	x_1	x_2	s_1	s_2	s_3	Solution
z	-1	3	2	0	0	0	0
s_1	0	-3	-1	1	0	0	-3
s_2	0	-4	-3	0	1	0	-6
s_3	0	1	1	0	0	1	3

Var. en base	z	x_1	x_2	s_1	s_2	s_3	Solution
z	-1	$\frac{1}{3}$	0	0	$\frac{2}{3}$	0	-4
s_1	0	$-\frac{5}{3}$	0	1	$-\frac{2}{3}$	0	-1
x_2	0	$\frac{4}{3}$	1	0	$-\frac{1}{3}$	0	2
s_3	0	$-\frac{1}{3}$	0	0	$\frac{1}{3}$	1	1

Var. en base	z	x_1	x_2	s_1	s_2	s_3	Solution
z	-1	0	0	$\frac{1}{5}$	$\frac{3}{5}$	0	$-\frac{21}{5}$
x_1	0	1	0	$-\frac{3}{5}$	$\frac{1}{5}$	0	$\frac{3}{5}$
x_2	0	0	1	$\frac{4}{5}$	$-\frac{3}{5}$	0	$\frac{6}{5}$
s_3	0	0	0	$-\frac{1}{5}$	$-\frac{2}{5}$	1	$\frac{6}{5}$

6 Solveurs et langages de modélisation

Exemple 17 (Production de jouets).

- Une société de jouets produit des trains, des camions et des voitures, en utilisant 3 machines.
- Les disponibilités quotidiennes des 3 machines sont 430, 460 et 420 minutes, et les profits par train, camion et voiture sont respectivement EUR 3, EUR 2 et EUR 5.
- Les temps nécessaires sur chaque machine sont :

Machine	Train	Camion	Voiture
1	1	2	1
2	3	0	2
3	1	4	0

Primal

$$\begin{aligned}
 \max z = & 3x_1 + 2x_2 + 5x_3 \\
 \text{s.c.} \quad & x_1 + 2x_2 + x_3 \leq 430 \\
 & 3x_1 + x_3 \leq 460 \\
 & x_1 + 4x_2 \leq 420 \\
 & x_1, x_2, x_3 \geq 0
 \end{aligned}$$

$$\begin{aligned}
 x_1 &= 0 \\
 x_2 &= 100 \\
 x_3 &= 230 \\
 z &= 1350
 \end{aligned}$$

Dual

$$\begin{aligned}
 \min w = & 430y_1 + 460y_2 + 420y_3 \\
 \text{s.c.} \quad & y_1 + 3y_2 + y_3 \geq 3 \\
 & 2y_1 + 4y_3 \geq 2 \\
 & y_1 + 2y_2 \geq 5 \\
 & y_1, y_2, y_3 \geq 0
 \end{aligned}$$

$$\begin{aligned}
 y_1 &= 1 \\
 y_2 &= 2 \\
 y_3 &= 0
 \end{aligned}$$

Var. en base	z	x_1	x_2	x_3	s_1	s_2	s_3	Solution
z	-1	-4	0	0	-1	-2	0	-1350
x_2	0	$-\frac{1}{4}$	1	0	$\frac{1}{2}$	$-\frac{1}{4}$	0	100
x_3	0	$\frac{3}{2}$	0	1	0	$\frac{1}{2}$	0	230
s_3	0	2	0	0	-2	1	1	20

$$\text{Base : } B = \{x_2, x_3, s_3\}.$$

Solveurs

Logiciels pour résoudre des programmes linéaires :

- Indépendants :
 - Commerciaux : CPLEX (IBM), XPRESS-MP (<http://www.fico.com/>), Gurobi (<http://www.gurobi.com/>), ...
 - Gratuits : PCx, lpsolve, glpk, ...
- Tableurs : La plupart des tableurs intègrent un outil de résolution de programmes linéaires (Excel, Gnumeric, ...)
- Langages de modélisation (ampl, GNU MathProg, mpl, OPL studio, mosel, ...) : langages de haut niveau permettant la séparation modèle/données, se chargeant de l'interface avec un solveur.

NAME	toys		
ROWS			
L	R0001		
L	R0002		
L	R0003		
N	R0004		
COLUMNS			
C0001	R0001	1	
C0001	R0002	3	
C0001	R0003	1	
C0001	R0004	3	
C0002	R0001	2	
C0002	R0003	4	
C0002	R0004	2	
C0003	R0001	1	
C0003	R0002	2	
C0003	R0004	5	
RHS			
B	R0001	430	
B	R0002	460	
B	R0003	420	
ENDATA			

FIGURE 1 – Exemple de production de jouets au format MPS

Solveurs indépendants

Avantages

- Puissance, efficacité
- Intégrables dans des applications via des librairies

Désavantages

- Formats de fichiers (MPS)
- Pas de séparation modèle / données
- Ré-utilisation difficile des modèles

Solveurs intégrés aux tableurs

Avantages

- Disponibles sur (quasi) tous les ordinateurs
- Interface facile d'utilisation
- Présentation des données / résultats

Désavantages

- Difficulté d'implémenter de grands modèles
- Séparation modèle / données difficile
- Solveurs moins efficaces (en général)

Langages de modélisation

Avantages

- Séparation modèle / données
- Ré-utilisabilité des modèles
- Indépendance modèle / solveur

```

#
# Data definition
#

set Toys;
param nMachines;
set Machines := 1..nMachines;

param profit {Toys};
param time {Machines,Toys};
param avail {Machines};

#
# Variables
#

var prod {Toys} >= 0;

#
# Objective
#

maximize total_profit:
sum{t in Toys} profit[t]*prod[t];

#
# Constraints
#

subject to machine_usage {m in Machines}:
sum{t in Toys} time[m,t] * prod[t] <= avail[m];

```

FIGURE 2 – Exemple de production de jouets au format AMPL (modèle)

Désavantages

- Apprentissage du langage
- Prix des versions commerciales
- Limitation en taille des versions d’essai gratuites
- Moindre efficacité (actuellement) des solveurs gratuits

```

#
# Data
#

data;

set Toys := Trains, Trucks, Cars ;
param nMachines := 3;

param profit :=
Trains 3
Trucks 2
Cars 5 ;

param time : Trains Trucks Cars :=
1 1 2 1
2 3 0 2
3 1 4 0 ;

param avail :=
1 430
2 460
3 420 ;

end;

```

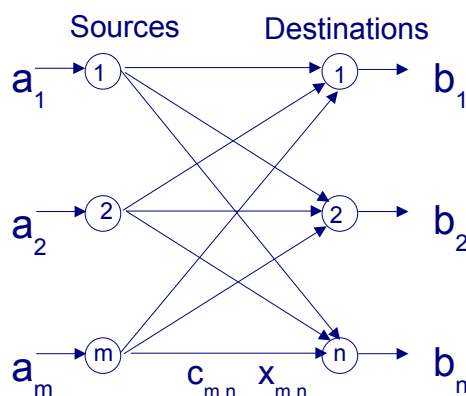
FIGURE 3 – Exemple de production de jouets au format AMPL (données)

Troisième partie

Le problème de transport

7 Définition et exemples

- Un produit doit être transporté de *sources* (usines) vers des *destinations* (dépôts, clients).
- *Objectif* : déterminer la quantité envoyée de chaque source à chaque destination en minimisant les coûts de transport. Les coûts sont proportionnels aux quantités transportées.
- *Contraintes* d'offre limitée aux sources et de demande à satisfaire aux destinations.



Exemple 18 (Modèle de transport).

- Une firme automobile a trois usines à Los Angeles, Detroit et New Orleans, et deux centres de distribution à Denver et Miami.
- Les capacités des trois usines sont de 1000, 1500 et 1200 respectivement, et les demandes aux centres de distribution sont de 2300 et 1400 voitures.
- Coûts :

	Denver	Miami
Los Angeles	80	215
Detroit	100	108
New Orleans	102	68

Formulation

$$\begin{aligned}
 \min z = & 80x_{11} + 215x_{12} + 100x_{21} + 108x_{22} + 102x_{31} + 68x_{32} \\
 \text{s.c.} & \\
 (\text{Los Angeles}) & x_{11} + x_{12} = 1000 \\
 (\text{Detroit}) & x_{21} + x_{22} = 1500 \\
 (\text{New Orleans}) & x_{31} + x_{32} = 1200 \\
 (\text{Denver}) & x_{11} + x_{21} + x_{31} = 2300 \\
 (\text{Miami}) & x_{12} + x_{22} + x_{32} = 1400 \\
 & x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32} \geq 0
 \end{aligned}$$

Représentation tableau

	Denver	Miami	Offre
Los Angeles	80 1000	215	1000
Detroit	100 1300	108 200	1500
New Orleans	102	68 1200	1200
Demande	2300	1400	

Problèmes non balancés

- Si l'offre n'est pas égale à la demande : modèle non balancé.
- Introduction d'une source ou destination artificielle.

	Denver	Miami	Offre
Los Angeles	80 1000	215	1000
Detroit	100 1300	108	1300
New Orleans	102	68 1200	1200
Artif.	0	0 200	200
Demande	2300	1400	

Variantes

Le modèle de transport n'est pas limité au transport de produits entre des sources et destinations géographiques.

Exemple 19 (Modèle de production).

- Une société fabrique des sacs à dos, pour lesquels la demande arrive de mars à juin et est de 100, 200, 180 et 300 unités, respectivement.
- La production pour ces mois est de 50, 180, 280 et 270, respectivement.
- La demande peut être satisfaite
 1. par la production du mois courant (\$40 / sac) ;
 2. par la production d'un mois précédent (+ \$0.5 / sac / mois pour le stockage) ;
 3. par la production d'un mois suivant (+ \$2 / sac / mois de pénalité de retard).

Correspondances avec le modèle de transport

Transport	Production - stocks
Source i	Période de production i
Destination j	Période de demande j
Offre à la source i	Capacité de production à la période i
Demande à la destination j	Demande pour la période j
Coût de transport de i à j	Coût unitaire (production + stock + pénalité) pour une production en période i pour la période j

8 Algorithme pour le problème de transport

Algorithme pour le problème de transport

Basé sur l'algorithme du simplexe en tenant compte de la structure du problème.

1. Détermination d'une solution de base admissible.
2. Détermination de la variable entrant en base.

3. Détermination de la variable sortant de base.

Exemple 20 (Algorithme pour le problème de transport).

	1	2	3	4	Offre
1	10	2	20	11	15
2	12	7	9	20	25
3	4	14	16	18	10
Demande	5	15	15	15	

Détermination d'une solution de base admissible

- Heuristiques "gloutonnes", pas besoin de méthode des deux phases.
- Variantes :
 1. Coin Nord-Ouest
 2. Méthode des moindres coûts

Coin Nord-Ouest

Partir du coin supérieur gauche du tableau.

1. allouer le plus possible à la cellule courante et ajuster l'offre et la demande ;
2. se déplacer d'une cellule vers la droite (demande nulle) ou le bas (offre nulle) ;
3. répéter jusqu'au moment où toute l'offre est allouée.

Exemple 21 (Coin Nord-Ouest).

	1	2	3	4	Offre
1	10	2	20	11	15
2	12	7	9	20	25
3	4	14	16	18	10
Demande	5	15	15	15	

Coût : 520

Moindres coûts

Sélectionner la cellule de coût minimum.

1. allouer le plus possible à la cellule courante et ajuster l'offre et la demande ;
2. sélectionner la cellule de coût minimum ayant une demande et une offre non nulles ;
3. répéter jusqu'au moment où toute l'offre est allouée.

Exemple 22 (Moindres coûts).

	1	2	3	4	Offre
1	10	2	20	11	15
2	12	7	9	20	25
3	4	14	16	18	10
Demande	5	15	15	15	

Coût : 475

Formulation

$$\begin{aligned}
 \min z &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\
 \text{s.c. } (u_i) \quad &\sum_{j=1}^n x_{ij} = a_i \quad i = 1, \dots, m \\
 (v_j) \quad &\sum_{i=1}^m x_{ij} = b_j \quad j = 1, \dots, n \\
 &x_{ij} \geq 0 \quad i = 1, \dots, m, j = 1, \dots, n
 \end{aligned}$$

Problème dual

$$\begin{aligned}
 \max w &= \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \\
 \text{s.c. } \quad &u_i + v_j \leq c_{ij} \quad i = 1, \dots, m, j = 1, \dots, n
 \end{aligned}$$

Adaptation du simplexe

Critère d'optimalité :

$$u_i + v_j - c_{ij} \leq 0$$

Complémentarité :

$$x_{ij} > 0 \Rightarrow u_i + v_j - c_{ij} = 0$$

Trois étapes :

1. détermination des variables duales (multiplicateurs);
2. vérification du critère d'optimalité et détermination de la variable entrante;
3. détermination de la variable sortante pour préserver l'admissibilité et pivotage.

Détermination des variables duales

1. $m + n - 1$ équations à $m + n$ inconnues : fixer $u_1 = 0$.
2. Résoudre récursivement le système

$$u_i + v_j - c_{ij} = 0 \text{ pour tout } x_{ij} > 0.$$

Exemple 23 (Détermination des variables duales).

	1	2	3	4	Offre
1	10	2	20	11	15
	5	10			
2	12	7	9	20	25
		5	15	5	
3	4	14	16	18	10
			10		
Demande	5	15	15	15	

$$\begin{aligned}
 &u_1 = 0 \\
 u_1 + v_1 = 10 &\Rightarrow v_1 = 10 \\
 u_1 + v_2 = 2 &\Rightarrow v_2 = 2 \\
 u_2 + v_2 = 7 &\Rightarrow u_2 = 5 \\
 u_2 + v_3 = 9 &\Rightarrow v_3 = 4 \\
 u_2 + v_4 = 20 &\Rightarrow v_4 = 15 \\
 u_3 + v_4 = 18 &\Rightarrow u_3 = 3
 \end{aligned}$$

Vérification du critère d'optimalité et détermination de la variable entrante

	1	10	2	2	3	4	4	15	Offre
1	10		2		20		11		15
0	5		10		-16		4		
2	12		7		9		20		25
5		3	5		15		5		
3	4		14		16		18		10
3		9		-9		-9	10		
Demande	5		15		15		15		

Détermination de la variable sortante pour préserver l'admissibilité et pivotage

Objectifs : 1. l'offre et la demande doivent continuer à être satisfaites ;

2. les quantités transportées doivent rester positives.

Méthode : 1. construction d'un *cycle* parcourant des variables en base en partant de et revenant à la variable entrante ;

2. déplacement le long de lignes et colonnes en alternant ajout et retrait d'une même quantité.

	1	10	2	2	3	4	4	15	Offre
1	10	$-\theta$	2	$+\theta$	20		11		15
0	5		10		-16		4		
2	12		7	$-\theta$	9		20	$+\theta$	25
5		3	5		15		5		
3	4	$+\theta$	14		16		18	$-\theta$	10
3		9		-9		-9	10		
Demande	5		15		15		15		

$$\theta = 5$$

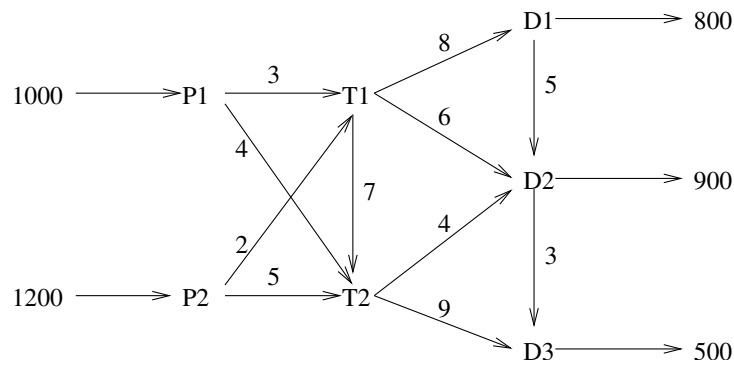
	1	1	2	2	3	4	4	15	Offre
1	10		2	$-\theta$	20		11	$+\theta$	15
0		-9	15		-16		4		
2	12		7	$+\theta$	9		20	$-\theta$	25
5		-6	0		15		10		
3	4		14		16		18		10
3	5			-9		-9	5		
Demande	5		15		15		15		

$$\theta = 10$$

	1	-3	2	2	3	4	4	11	Offre
1	10		2		20		11		15
0		-13	5		-16		10		
2	12		7		9		20		25
5		-10	10		15			-4	
3	4		14		16		18		10
7	5			-5		-5	5		
Demande	5		15		15		15		

9 Le problème de transbordement

- Extension du modèle de transport : il est parfois nécessaire (ou moins cher) d'utiliser des *noeuds intermédiaires* pour le transport.
- Deux usines $P1$ et $P2$ servent 3 vendeurs $D1$, $D2$ et $D3$, via deux centres de transit $T1$ et $T2$.



Transformation en problème de transport

— 3 types de noeuds :

Noeuds d'offre purs : arcs sortants uniquement. *offre = offre originale*

Noeuds de demande purs : arcs entrants uniquement. *demande = demande originale*

Noeuds de transbordement : arcs entrants et sortants. *offre/demande = offre/demande originale + buffer*

— Les noeuds de transbordement sont à la fois sources et destinations pour le problème de transport.

— *Buffer* : quantité nécessaire pour transporter toute la demande à travers le noeud de transbordement.

— Dans notre exemple : $B = 2200$.

	T1	T2	D1	D2	D3	Offre
P1	3	4	M	M	M	1000
P2	2	5	M	M	M	1200
T1	0	7	8	6	M	2200
T2	M	0	M	4	9	2200
D1	M	M	0	5	M	2200
D2	M	M	M	0	3	2200
Demande	2200	2200	3000	3100	500	

Quatrième partie

Le flot maximum

10 Introduction

- Le problème de flot maximal consiste à transporter la quantité maximale possible d'une origine (*source*) à une destination (*puits*) données, sans dépasser les capacités des arcs.
- Données : un graphe dirigé $G = (N, A)$, une capacité $u_{ij} > 0$ pour tout arc (i, j) , une source s et un puits t .

Définition 9 (*st-flot*). — Un *st-flot* dans le réseau est un vecteur d'entiers $f = (f_{ij})$ associés aux arcs tel que :

$$\begin{aligned} 0 \leq f_{ij} \leq u_{ij} & \quad \forall (i, j) \in A \\ \sum_{j:(i,j) \in A} f_{ij} = \sum_{j:(j,i) \in A} f_{ji} & \quad \forall i \in N \setminus \{s, t\} \end{aligned}$$

- La *valeur du flot* est

$$v(f) = \sum_{j:(s,j) \in A} f_{sj} = \sum_{j:(j,t) \in A} f_{jt}.$$

11 Formulation

Formulation du problème de flot maximal

$$\begin{aligned} \max \quad & v \\ \text{s.c.} \quad & \sum_{j:(i,j) \in A} f_{ij} - \sum_{j:(j,i) \in A} f_{ji} = \begin{cases} v & i = s, \\ -v & i = t, \\ 0 & \text{sinon,} \end{cases} \quad i \in N, \\ & 0 \leq f_{ij} \leq u_{ij} \quad (i, j) \in A. \end{aligned}$$

12 Coupe minimum

Définition 10 (*Coupe*). — La *coupe* dans le graphe $G = (N, A)$, définie par le sous-ensemble non vide de noeuds S , est l'ensemble d'arcs noté $\delta^+(S)$ sortant de S .

$$\delta^+(S) = \{(i, j) \in A : i \in S, j \in N \setminus S\}.$$

- Une *st-coupe* est une coupe qui sépare s et t , i.e. telle que $s \in S$ et $t \in N \setminus S$.
- La *capacité* d'une coupe est la somme des capacités de ses arcs.

Théorème 8 (Dualité faible). La capacité d'une *st-coupe* est supérieure ou égale à la valeur d'un *st-flot*.

Théorème 9 (Dualité forte). La valeur d'un *st-flot* maximal est égale à la capacité d'une *st-coupe* minimale.

13 Algorithme de Ford-Fulkerson

Définition 11 (Arcs en avant / en arrière). Soit f un *st-flot*, et P un chemin dans le graphe non dirigé obtenu en remplaçant chaque arc de A par une arête. Un arc traversé dans P dans son orientation originale est appelé *arc en avant*, un arc traversé dans l'autre sens est appelé *arc en arrière*.

Définition 12 (Chemin augmentant). Si pour tout arc (i, j) en avant de P , $f_{ij} < u_{ij}$ et pour tout arc (i, j) en arrière de P , $f_{ji} > 0$, alors P est appelé chemin *f-augmentant*.

Théorème 10 (Critère d'optimalité). Un *st-flot* f est maximal si et seulement si il n'existe pas de chemin *f-augmentant*.

Algorithme de Ford et Fulkerson

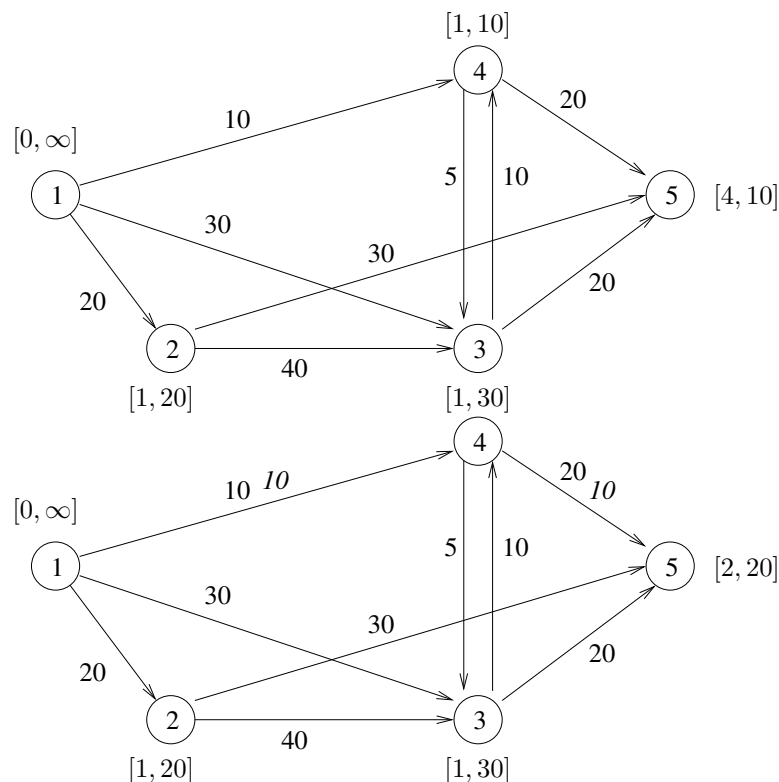
- Idée de l'algorithme : trouver un chemin augmentant et augmenter le flot sur ce chemin.
 - Initialisation : $f = 0$.
 - Alternance de deux phases :
 1. *Phase de marquage* (recherche d'un chemin augmentant).
 2. *Phase d'augmentation* (augmenter le flot sur le chemin trouvé en phase de marquage).
- Ces deux phases sont répétées jusqu'au moment où il n'existe plus de chemin augmentant.

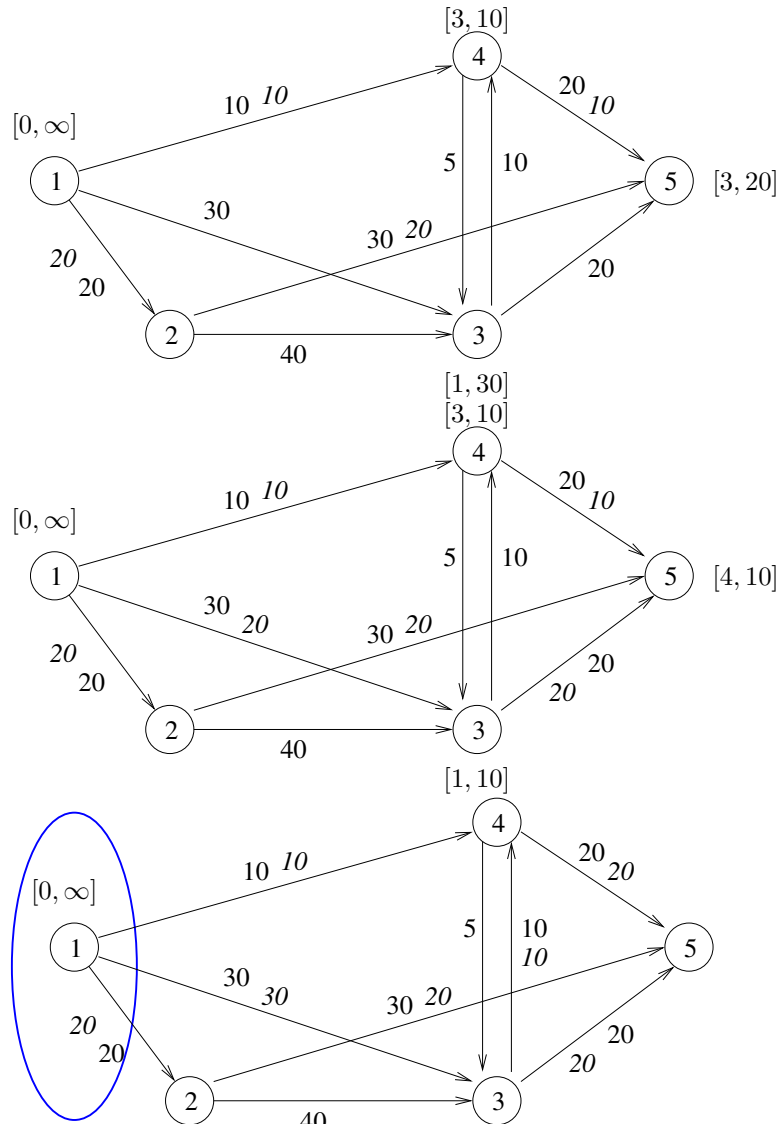
Phase de marquage

1. Marquer s par $[0, \infty]$, $L = \{s\}$.
2. Tant que $L \neq \emptyset$ et t non marqué :
 - (a) Sélectionner i dans L et le retirer de L .
 - (b) Pour tout j non marqué tel que $(i, j) \in A$ et $f_{ij} < u_{ij}$, marquer j par $[i, \alpha_j]$ avec $\alpha_j = \min(\alpha_i, u_{ij} - f_{ij})$ et ajouter j dans L .
 - (c) Pour tout j non marqué tel que $(j, i) \in A$ et $f_{ji} > 0$, marquer j par $[i, \alpha_j]$ avec $\alpha_j = \min(\alpha_i, f_{ji})$ et ajouter j dans L .
3. Si t est non marqué, STOP (plus de chemin augmentant).

Phase d'augmentation

1. $j = t$.
2. Tant que $j \neq s$:
 - Soit $[i, \alpha_j]$ la marque de j . Si (i, j) est en avant, $f_{ij} = f_{ij} + \alpha_j$. Si (i, j) est en arrière, $f_{ji} = f_{ji} - \alpha_j$.
 - $j = i$.





Les noeuds marqués à la dernière itération définissent la coupe minimale

Complexité

- Marquage : $O(|N| + |A|)$.
- Suppression des marques : $O(|N|)$.
- Examen de tous les successeurs et tous les prédécesseurs : $O(|A|)$.
- Augmentation : $O(|N|)$.
- Après l'étape 2, retour à l'étape 1. A chaque itération on augmente le flot d'une unité au moins.
- Si la valeur de ce flot maximal est F , la complexité maximale totale est en $O(F \cdot (|N| + |N| + |A|)) = O(F \cdot (|N| + |A|))$.
- L'algorithme d'Edmonds et Karp a une complexité en temps qui ne dépend pas de la valeur du flot maximum.

Cinquième partie

Flots de coût minimum

14 Définitions et exemples

- Extension du problème de transbordement : ajout de capacités sur les arcs.
- Données : un graphe dirigé $G = (N, A)$, une capacité $u_{ij} > 0$ et un coût $c_{ij} > 0$ pour tout arc (i, j) , une offre b_i pour tout noeud i (si $b_i < 0$, demande).
- La somme des offres/demandes doit être nulle.

Définition 13 (Flot compatible). Un *flot compatible* avec l'offre b est un vecteur d'entiers $f = (f_{ij})$ associés aux arcs tel que :

$$\begin{aligned} 0 \leq f_{ij} \leq u_{ij} & \quad (i, j) \in A \\ \sum_{j:(i,j) \in A} f_{ij} - \sum_{j:(j,i) \in A} f_{ji} = b_i & \quad i \in N \end{aligned}$$

But : Trouver un flot f compatible avec l'offre b de coût total

$$\sum_{(i,j) \in A} c_{ij} f_{ij}$$

minimal.

Bornes inférieures

- Dans certains cas, une borne inférieure sur les flots peut être donnée (quantité minimale présente sur un arc).

$$\begin{aligned} l_{ij} \leq f_{ij} \leq u_{ij} & \quad (i, j) \in A \\ \sum_{j:(i,j) \in A} f_{ij} - \sum_{j:(j,i) \in A} f_{ji} = b_i & \quad i \in N \end{aligned}$$

- Transformation :

$$f'_{ij} = f_{ij} - l_{ij}, \quad u'_{ij} = u_{ij} - l_{ij}, \quad b'_i = b_i - l_{ij}, \quad b'_j = b_j + l_{ij}$$

Techniques de résolution

1. Transformation en problème de *st*-flot et algorithme de chemins augmentants.
2. Formulation comme un programme linéaire et application particulière du simplexe (simplexe réseau).
 - La première méthode est la plus efficace pour des problèmes de flots "purs".
 - La deuxième méthode peut être appliquée quand le réseau apparaît comme sous-structure d'un programme linéaire plus général.

15 Algorithme de chemins augmentants

Transformation en problème de *st*-flot

1. ajouter une source s et un puits t artificiels ;
2. pour tout noeud i tel que $b_i > 0$, ajouter un arc (s, i) de coût nul et de capacité b_i ;
3. pour tout noeud i tel que $b_i < 0$, ajouter un arc (i, t) de coût nul et de capacité $-b_i$.

Le problème devient :

Trouver un *st*-flot de coût minimum et de valeur $B = \sum_{i:b_i > 0} b_i$.

Définition 14 (Graphe résiduel). Etant donné un flot f , le graphe résiduel correspondant est le graphe $G' = (N, A')$ défini par :

$$A' = \{(i, j) : (i, j) \in A \text{ et } f_{ij} < u_{ij}\} \\ \cup \{(j, i) : (i, j) \in A \text{ et } f_{ij} > 0\}$$

Pour déterminer les coûts réduits, un potentiel w_i est associé à chaque noeud i (variable duale).

Définition 15 (Coûts réduits). Etant donnés des potentiels w_i , $i \in V$, le coût réduit d'un arc dans le graphe résiduel est

$$c'_{ij} = c_{ij} - w_i + w_j \quad \text{si } (i, j) \in A, \\ c'_{ij} = -c_{ji} - w_i + w_j \quad \text{si } (j, i) \in A.$$

Idées de base

- Le st -flot à trouver est un flot maximal.
- Construction par l'augmentation du flot sur des chemins augmentants.
- Le chemin choisi dans le graphe résiduel doit être de coût réduit minimal (de manière à augmenter le flot à coût minimum).
- Au début de l'algorithme, le graphe résiduel est le graphe de départ et les potentiels sont tous nuls (donc les coûts réduits sont égaux aux coûts originaux).

Algorithme de chemins augmentants

1. Initialisation : $f = 0$, $w_i = 0$ pour tout noeud i .
2. Calculer le chemin augmentant de coût réduit minimal dans le graphe résiduel (e.g. Dijkstra).
3. Phase d'augmentation.
4. Mettre à jour les potentiels des noeuds, le graphe résiduel et les coûts réduits.

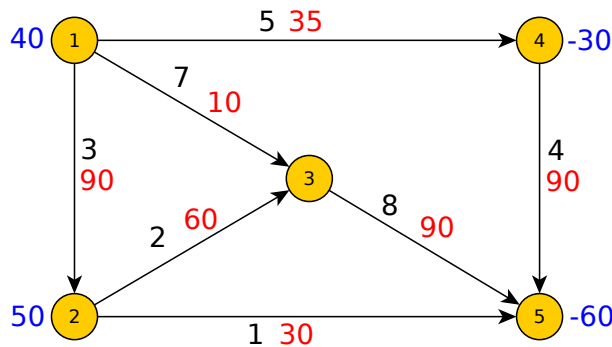
Répéter (2), (3) et (4) jusqu'au moment où toute l'offre est envoyée.

Mise à jour des potentiels

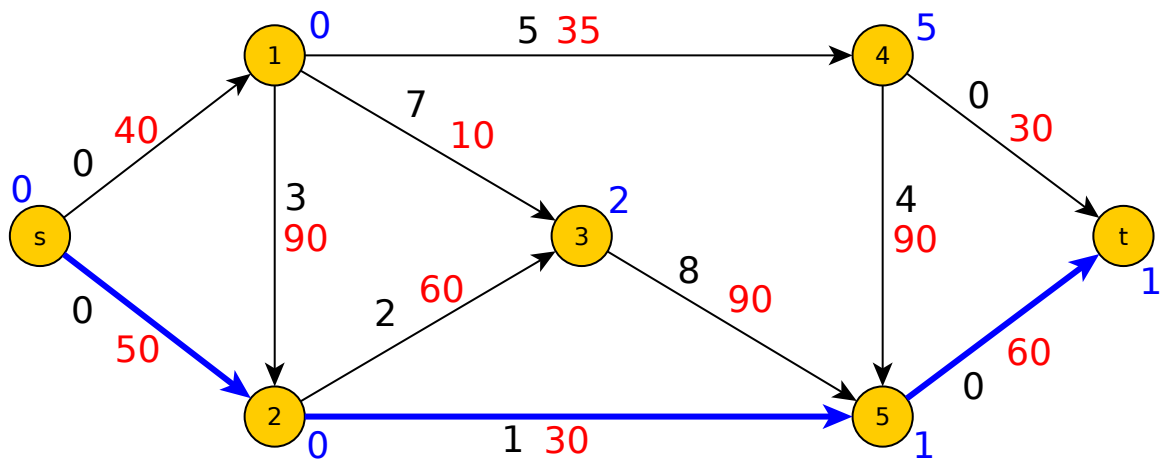
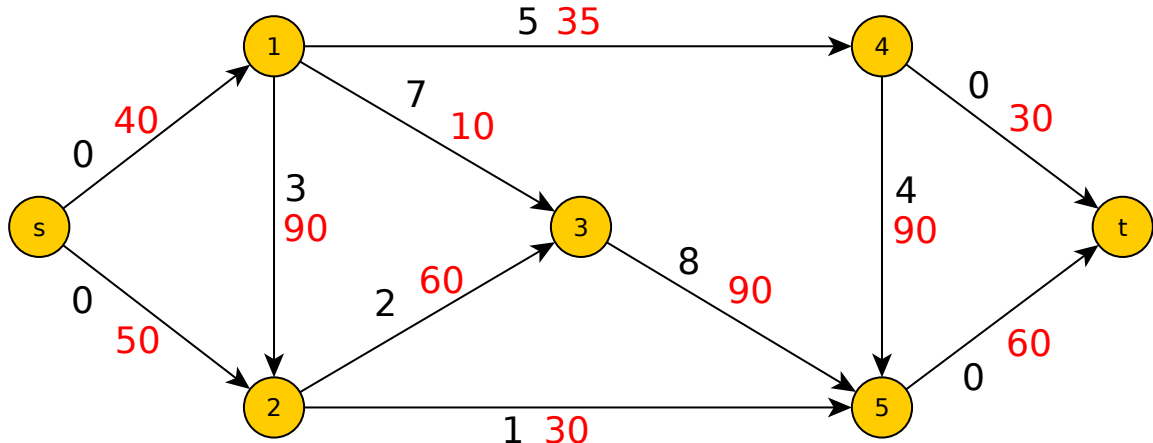
$$w_i = w_i - d(s, i) \quad \forall i \in N$$

où $d(s, i)$ est le coût réduit du chemin minimal de s à i dans le graphe résiduel.

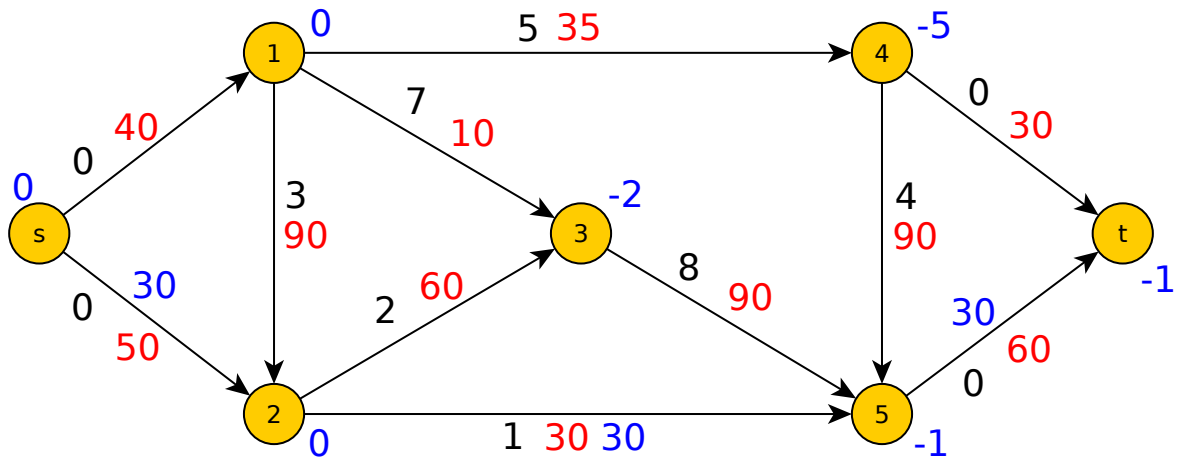
Exemple

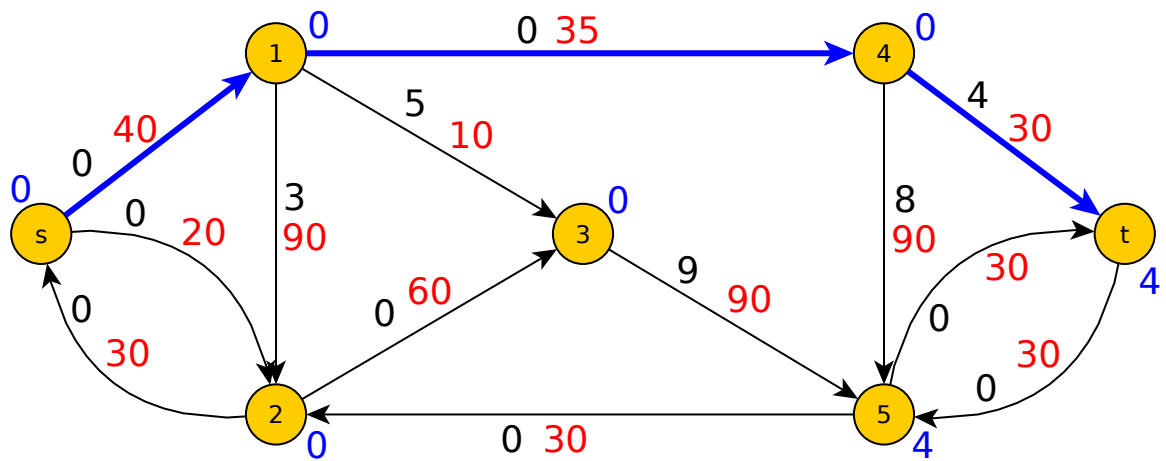


Itération 1

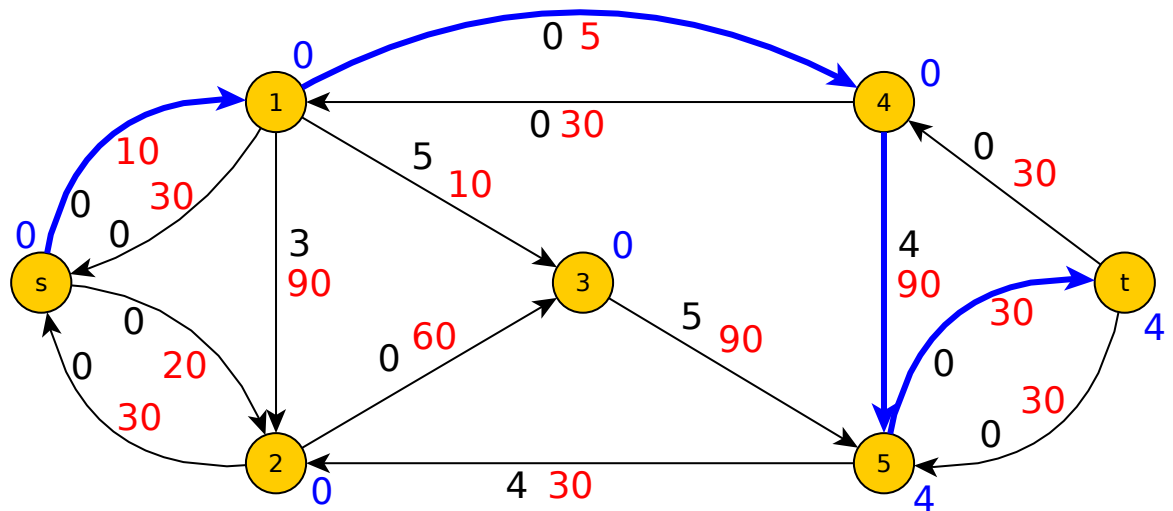
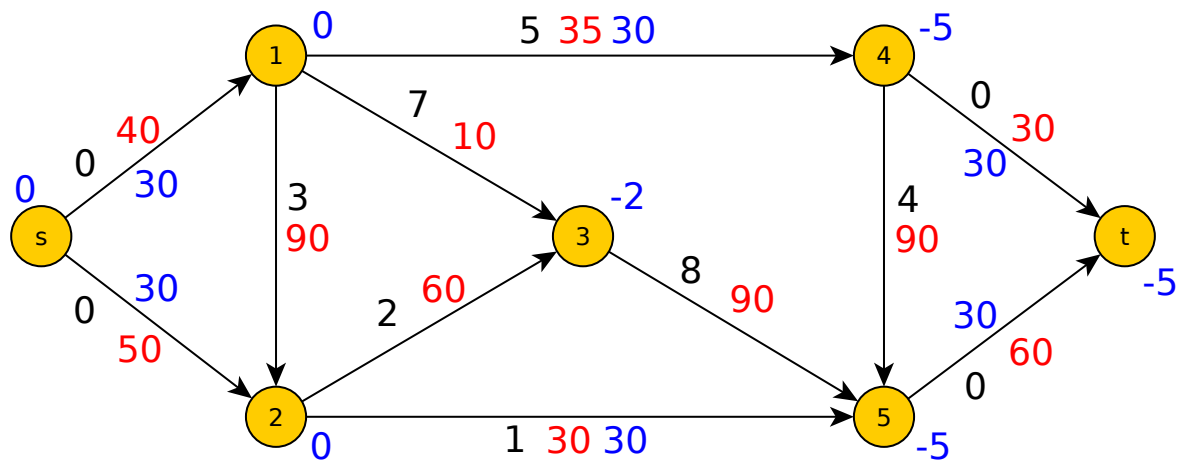


Itération 2

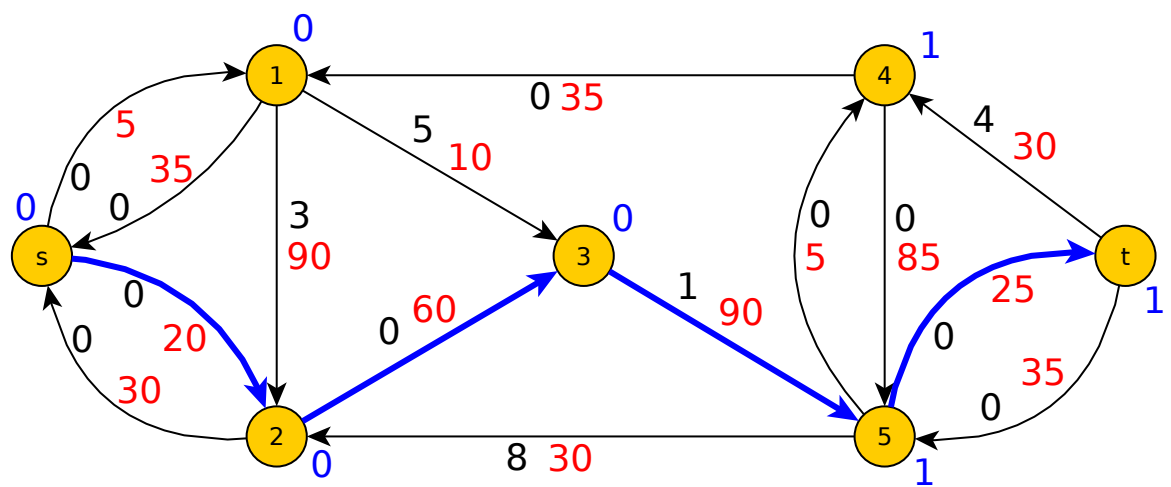
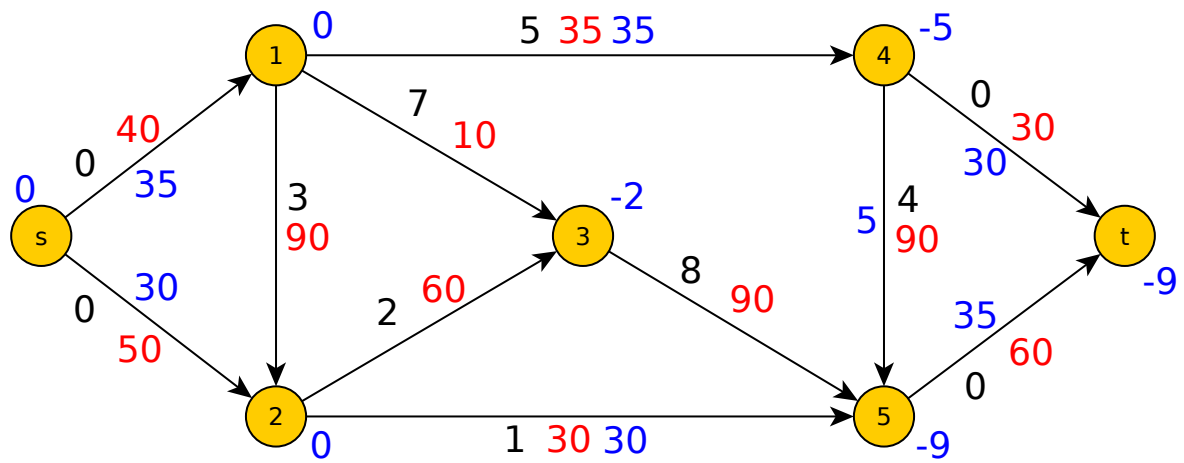




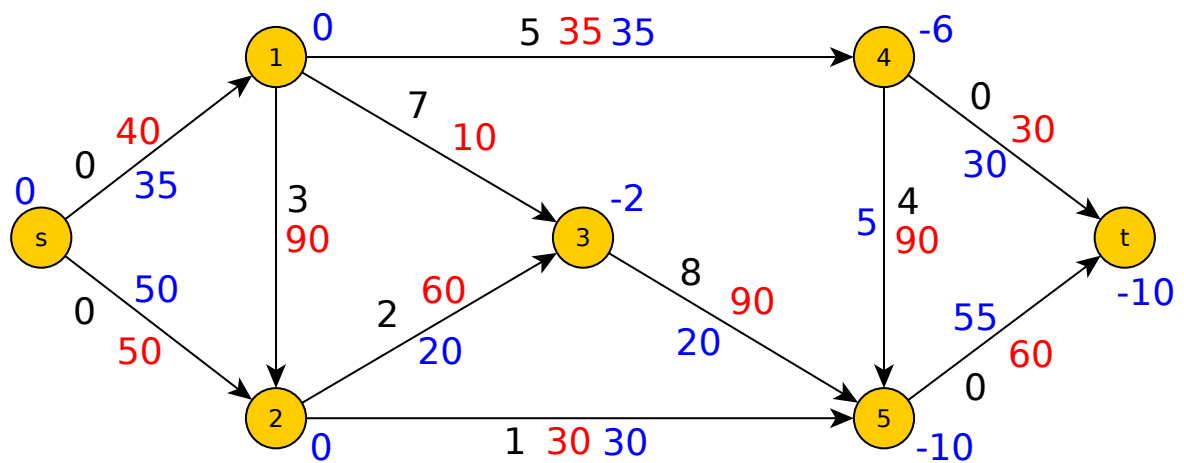
Itération 3

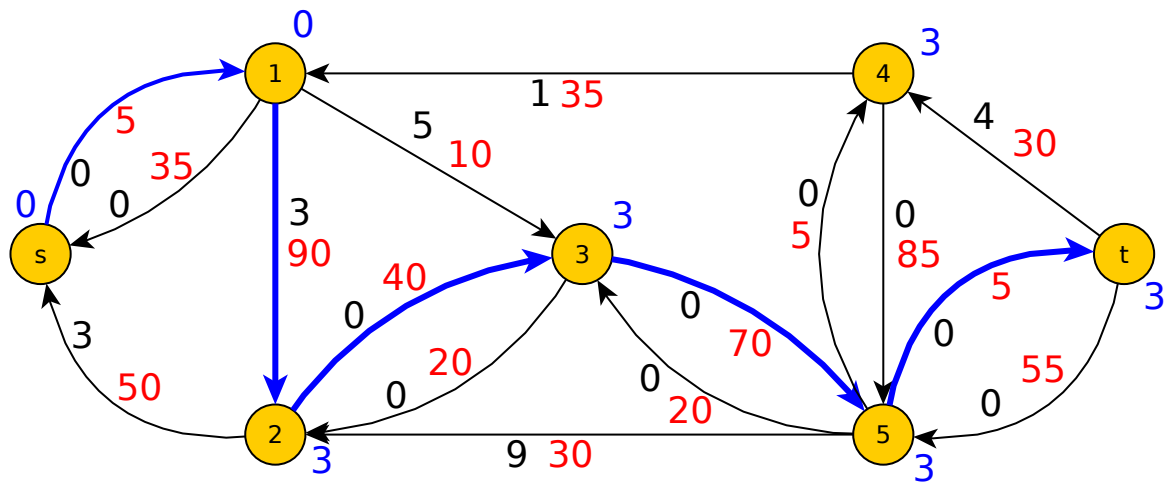


Itération 4

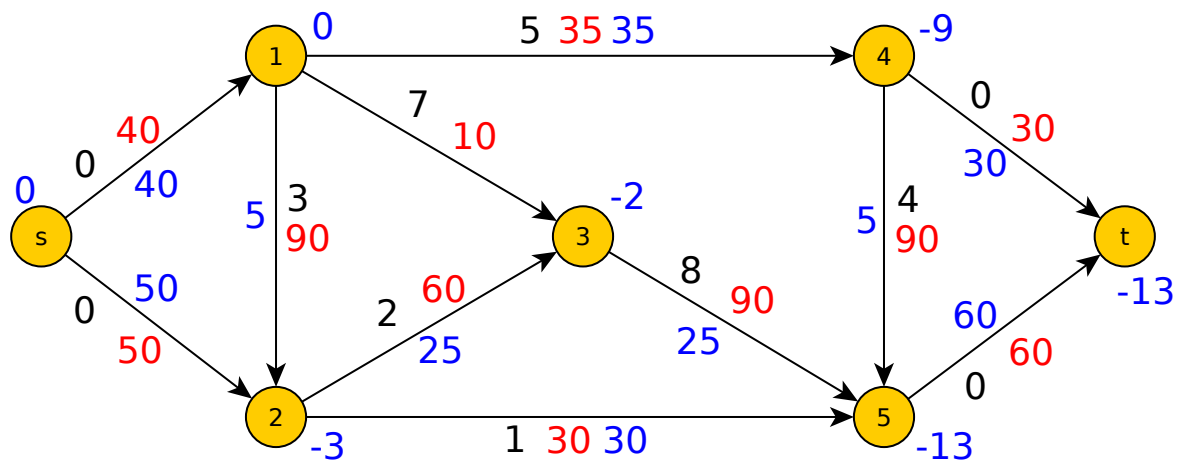


Itération 5





Solution finale



16 Extensions

Flots multi-produits

- Le réseau sert à transporter un ensemble K de produits. Chaque produit $k \in K$ a son propre vecteur de demande b^k . Les coûts peuvent également dépendre du produit (c_{ij}^k).
- La capacité est utilisée par tous les produits simultanément.

Formulation

$$\begin{aligned}
 \min \quad & \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k f_{ij}^k \\
 \text{s.c.} \quad & \sum_{j: (i,j) \in A} f_{ij}^k - \sum_{j: (j,i) \in A} f_{ji}^k = b_i^k \quad i \in N, k \in K, \\
 & \sum_{k \in K} f_{ij}^k \leq u_{ij} \quad (i,j) \in A, \\
 & f_{ij}^k \geq 0 \quad (i,j) \in A, k \in K.
 \end{aligned}$$

Problème de routage en télécommunications

- Cas particulier : application dans les réseaux de télécommunications.
- Chaque “produit” correspond à une paire de clients devant communiquer. Pour chaque produit, st -flot entre les deux clients s et t de valeur égale à la demande de ces clients.
- Programme linéaire de très grande taille (exemple : 100 noeuds, 400 arcs, 1000 demandes \Rightarrow 400 000 variables et 100 400 contraintes).
- Avec les solveurs actuels : quelques minutes de temps calcul.

Sixième partie

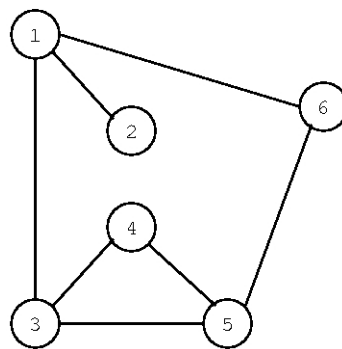
Couplages

17 Introduction

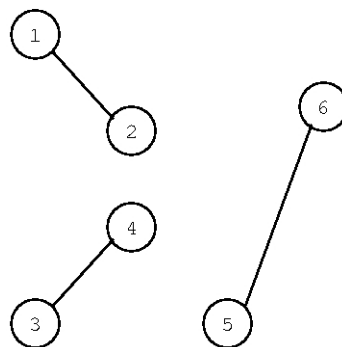
Étant donné un graphe non orienté, un couplage, ou matching en anglais, est un sous-ensemble d'arêtes sans sommet commun et dans lequel des sommets peuvent rester isolés (ces sommets n'ont alors aucune arête entrante ou sortante). Le couplage maximum est le couplage couvrant le plus grand nombre de sommets possibles, en laissant donc le moins de sommets isolés.

Pour illustrer cela, nous pouvons, par exemple, imaginer des jeunes universitaires diplômés indiquant les entreprises dans lesquelles ils désirent travailler, en les classant par ordre de préférence. D'un autre côté, les entreprises réalisent aussi un classement des jeunes diplômés qu'elles désirent préférentiellement accueillir. Le problème réside en la répartition des jeunes diplômés au sein des entreprises en satisfaisant au mieux les choix des jeunes diplômés et des entreprises. En termes de graphes, cet exemple revient donc à relier des sommets « jeunes diplômés » et des sommets « entreprises ». Néanmoins, cet exemple considère, en plus de la définition ci-dessus, un critère d'optimalité, puisque des préférences sont indiquées. Ainsi, ici, à chaque couplage doit aussi être associé une valeur indiquant la qualité du couplage à la lumière des préférences annoncées. Ce problème précis sera abordé à la section 19.

Considérons le graphe non dirigé suivant, où chaque sommet peut potentiellement être couplé avec n'importe quel autre sommet :

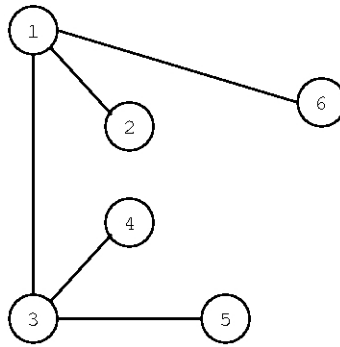


Nous pouvons « coupler » tous les sommets, par exemple, de la manière suivante :



Nous avons six sommets et trois couplages, il n'y a alors aucun sommet isolé, c'est une situation idéale.

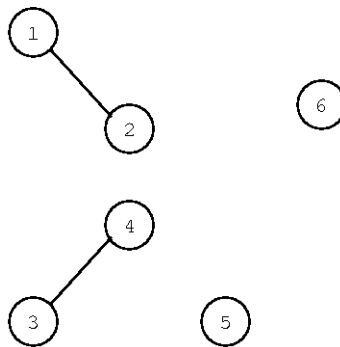
Par contre, considérons le graphe suivant :



Cette fois, nous ne pourrons coupler tous les sommets. Sinon, en effet, nous devons choisir un couplage reprenant le sommet 1. Si on choisit le couple $(1, 3)$, les sommets 2 et 6 sont isolés ; si on choisit le couple $(1, 2)$, le sommet 6 est isolé ; de même si on choisit le couple $(1, 6)$, le sommet 2 est alors isolé.

Il faut donc faire un choix tel que, sur l'ensemble du graphe, nous ayons le moins de sommets isolés. Ici, il apparaît qu'il faille éviter de coupler le sommet 1 avec le sommet 3, puisque cela a comme conséquence d'isoler deux sommets alors que le sommet 3 peut encore être couplé avec le sommet 4 ou le sommet 5.

Un exemple de couplage maximum est alors :



Nous avons alors 6 sommets et 2 couplages. D'autres couplages maxima existent, comme par exemple : $\{ (1, 6), (3, 4), 2, 5 \}$ ou $\{ (1, 6), (3, 5), 2, 4 \}$, etc.

18 Affectations bipartites

Souvent une telle technique de couplage s'applique sur des graphes bipartis. Un graphe biparti est, ici, un graphe non dirigé dans lequel toutes les arêtes relient des sommets appartenant à deux ensembles disjoints, aucune arête ne pouvant relier deux sommets d'un même ensemble.

Dans notre exemple de la section 17, les entreprises forment un ensemble et les jeunes diplômés en forment un second.

Il apparaît naturellement qu'un graphe biparti ne peut contenir de cycle de longueur impaire.

Le problème de l'affectation bipartite (bipartite assignment) consiste, étant donné un graphe biparti, à trouver un ensemble d'arcs de cardinalité maximum tel que chaque sommet est connecté à un autre sommet au plus ¹.

Considérer tous les couples possibles représente un trop grand travail. Une meilleure solution consiste à se baser sur la technique de flot maximum que nous avons vue au chapitre IV.

Pour construire une solution, on crée un sommet source dont les arcs sortants pointent, un à un, sur chaque élément d'un des deux ensembles de sommets du graphe biparti. Puis on ajoute un sommet puits vers lequel pointent tous les sommets de l'autre ensemble. Toutes les arêtes du réseau ainsi composé ont une capacité égale à 1. On applique alors l'algorithme de flot maximum sur ce réseau, en considérant que le sens des canalisations va toujours de la source au premier ensemble de sommets, puis vers le second ensemble de sommets, et enfin vers le puits.

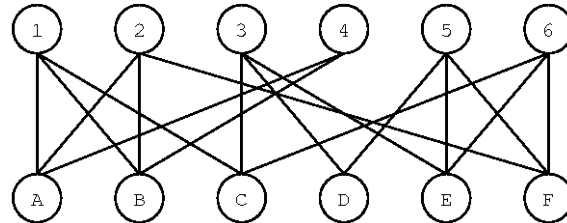
L'algorithme de flot donne un couplage biparti admissible car chaque sommet du premier ensemble de sommets est atteint par une unique arête de capacité égale à 1. Un tel sommet ne peut donc laisser passer, depuis la source,

1. Il s'agit donc d'un couplage maximum.

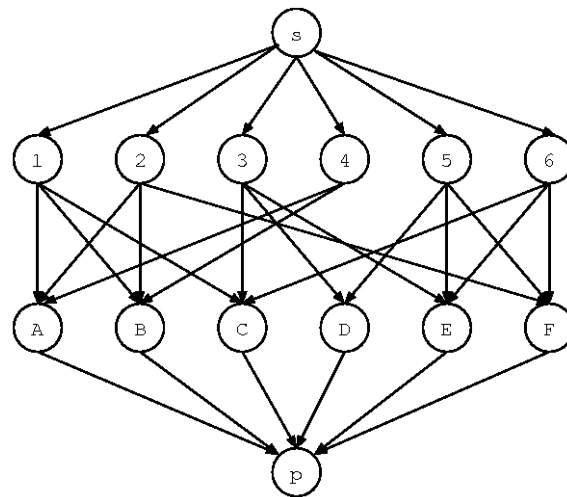
qu'une seule unité de flux. De même, chacun des sommets du second ensemble de sommets a une unique arête les reliant au puits. Eux aussi ne peuvent donc ne laisser passer, chacun, qu'une seule unité de flux. Ainsi, chaque sommet ne sera inclus qu'au plus une fois dans le couplage.

De plus, aucun autre couplage que celui produit par le flot maximum ne peut avoir plus d'arêtes, car cela conduirait à un flot supérieur à celui produit par le flot maximum.

Pour illustrer la réalisation d'un tel couplage, considérons le graphe biparti suivant :

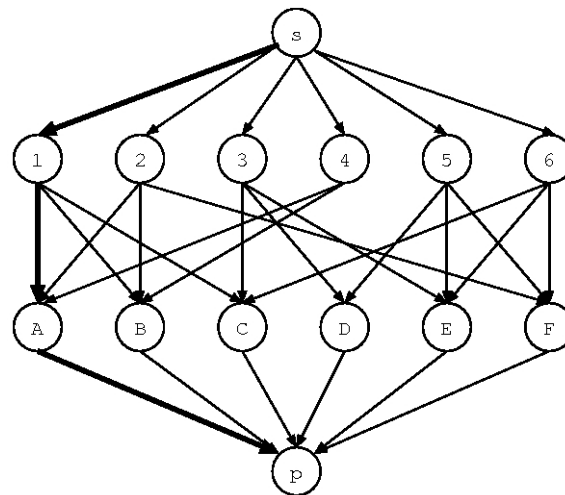


Pour transformer ce graphe biparti en réseau, nous y ajoutons une source et un puits. Nous considérons donc bien que toutes les capacités valent 1 :

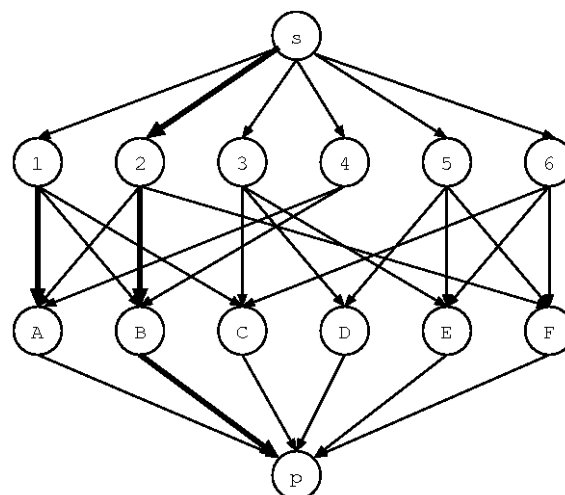


Appliquons à présent macroscopiquement l'algorithme de Ford et Fulkerson. Pour commencer, on identifie un premier chemin entre la source et le puits.

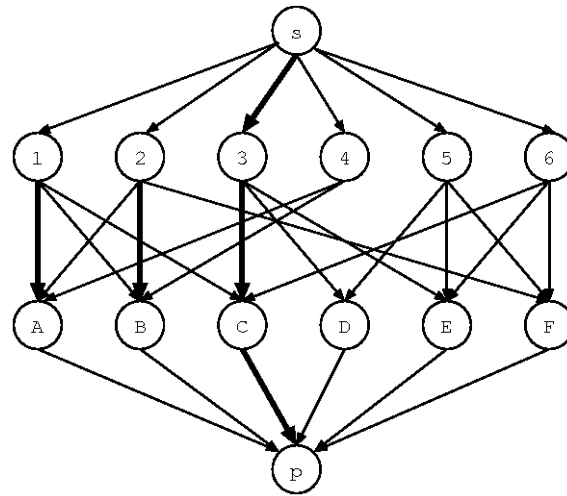
Prenons le chemin $s, 1, A$ et p :



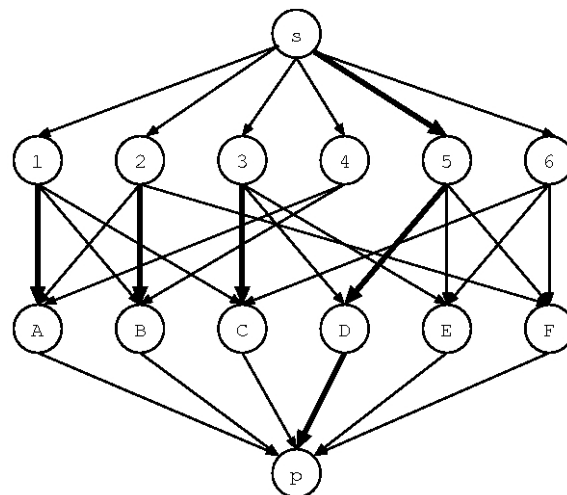
Nous pouvons à présent itérativement améliorer le flot en lui ajoutant la plus petite quantité de capacité non utilisée sur un arc faisant partie d'un chemin entre la source et le puits. Considérons donc, dans un premier temps, le chemin additionnel, $s, 2, B$ et p :



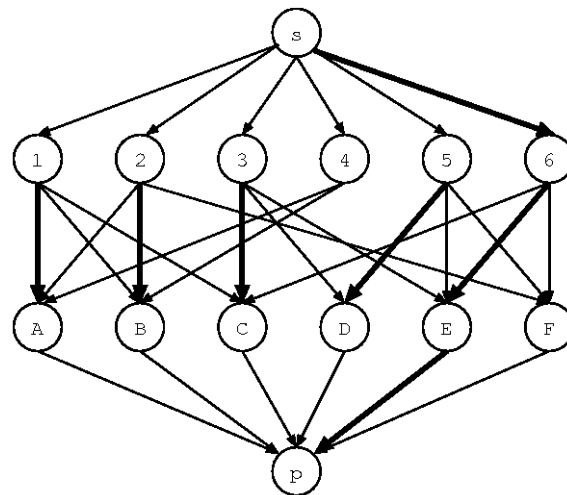
Il est encore possible d'augmenter le flot en suivant le chemin $s, 3, C$ et p :



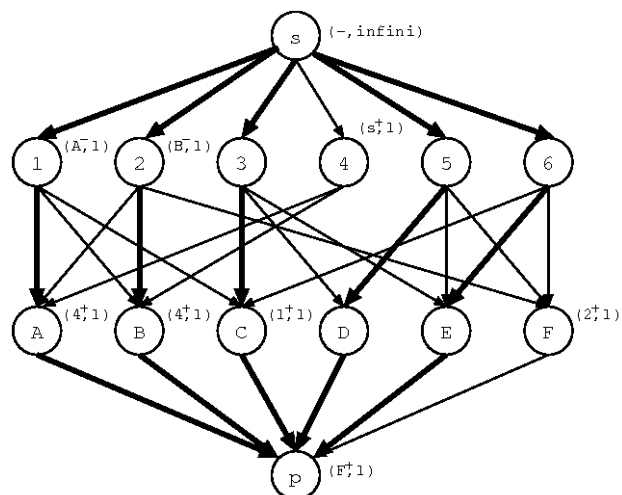
Nous ne pouvons augmenter le flot par un chemin partant de la source et allant au sommet 4, car tous les successeurs de ce sommet 4, à savoir les sommets A et B, sont déjà saturés. Nous prenons alors plutôt un chemin $s, 5, D$ et p :



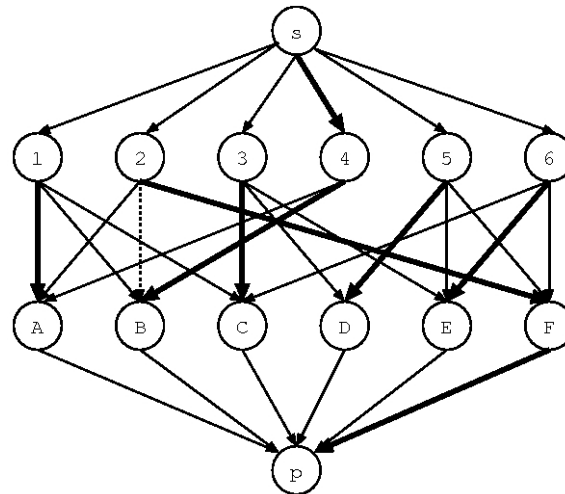
Il est toujours possible d'augmenter le flot en suivant le chemin $s, 6, E$ et p :



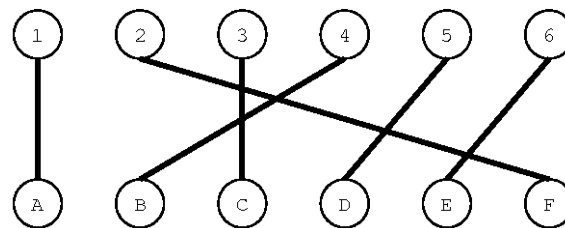
Les sommets 4 et F sont toujours isolés. A présent, pour augmenter encore le flot il est possible de récupérer le flot allant du sommet 2 au sommet B. Une partie de l'étiquetage est alors tel qu'indiqué ci-dessous, où les arcs en gras indiquent une canalisation saturée, ayant donc un flux y passant égal à la capacité :



Nous construisons le chemin $s, 4, B$, nous récupérons le flux vers le sommet 2, puis nous continuons vers le sommet F et enfin nous arrivons au sommet p :



A ce stade, nous ne pouvons plus augmenter le flot, le couplage maximal est atteint, produisant, sur notre graphe biparti, l'affectation bipartite recherchée :



Remarquons que cette technique ne gère pas la notion de préférences évoquée dans notre premier exemple impliquant les jeunes diplômés et les entreprises. Cet aspect est abordé à la section suivante.

19 Le problème des mariages stables

Le problème de l'affectation bipartite dans un digraphe biparti complètement interconnecté (un arc existe dans chaque sens pour chaque couple de sommets) où des préférences sont indiquées sur les arcs, peut correspondre au problème des jeunes diplômés et des entreprises évoqué au début de ce chapitre. Néanmoins, ce problème est classiquement évoqué comme le problème des mariages stables.

Considérons donc n femmes et n hommes exprimant chacun, sous forme d'une liste, leur préférence parmi l'ensemble des représentants de l'autre sexe (cette liste énumère toutes les personnes de l'autre sexe par ordre décroissant de préférence).

Le problème des mariages stables consiste à trouver un ensemble de n mariages maximisant la satisfaction de l'ensemble des hommes et femmes.

Un mariage sera dit instable si deux personnes qui ne sont pas mariés se préfèrent mutuellement à leur conjoint.

Le but consiste donc à chercher une configuration stable.

Une méthode possible de résolution consiste à faire en sorte que chaque homme, par exemple, se déclare à une femme (ou vice versa²).

Dans un premier temps, chaque homme se déclare à la première femme de sa liste de préférence.

— Si cet homme se déclare à une femme qui n'a pas encore de fiancé, ils se fiancent.

2. Nous allons néanmoins, puisque le problème original le décrit ainsi, considérer par la suite, le vieux système injuste où ce sont les hommes qui se déclarent aux femmes.

- Si celle-ci est déjà fiancée à un homme qu'elle préfère, le prétendant est éconduit et se déclare alors à la femme suivante sur sa liste de préférences, et ainsi de suite.
- Si cet homme se déclare à une femme fiancée mais qui le préfère lui, selon la liste de préférence de cette femme, à son fiancé, elle rompt ses fiançailles et se fiance avec notre homme. L'ex-fiancé à présent éconduit redevient un prétendant et se déclare à nouveau en continuant là où il en était dans sa liste de préférences.

Une fois tous les hommes et femmes fiancés, nous procédons aux différents mariages de nos couples ainsi formés. Cette méthode produit des mariages stables car une femme f qu'un homme h pourrait préférer à sa propre fiancée sera elle-même fiancée à quelqu'un qu'elle préfère à h .

Pour résoudre ce problème, nous utilisons une liste indiquant pour chaque femme le rang de chacun de ses prétendants. Le rang est le numéro d'ordre du prétendant dans la liste de préférences d'une femme. Ainsi, $\text{rang}[f][p]$ égale le rang du prétendant p sur la liste de préférences de la femme f . Pour chaque femme f , la valeur sentinelle $+\infty$ est mise en $\text{rang}[f][0]$ (ceci est nécessaire au bon fonctionnement de l'algorithme proposé ci-dessous).

Puisqu'un fiancé éconduit doit à nouveau se déclarer à la femme suivante de sa liste de préférences, nous devons mémoriser l'endroit où chaque prétendant est arrivé dans sa liste, nous utilisons un vecteur `suivante`, initialisé à zéro.

Nous manipulerons aussi une matrice `préférée[h][f]` qui indique la f^{me} femme dans la liste de préférences de l'homme h .

Enfin, le vecteur `fiancé` est tel que `fiancé[f]` indique le numéro du fiancé de la femme f , initialement ce vecteur n'est composé que de 0.

L'algorithme 1 réalise bien le processus détaillé ci-dessus. On considère chaque homme tour à tour comme un prétendant, et on le fiance à la première femme de sa liste de préférences qui est « libre » ou qui le préfère à son fiancé actuel. Si un tel fiancé est éconduit, il devient le prétendant et continue la recherche, dans sa liste de préférences, d'une femme à laquelle se fiancer.

Remarquons que lorsqu'un fiancé redevient un prétendant, donc lorsque nous exécutons les trois instructions du `if`, nous sommes dans la situation où le numéro de l'homme qui était fiancé et qui redevient prétendant est inférieur au numéro de l'homme h initialement considéré lors de ce tour de la boucle `for`.

Algorithme 1 Recherche des mariages stables

```
for(h=1 ; h <= n ; ++h)
{
    p=h
    while(p != 0)
    {
        ++suivante[p]
        f=préférée[p][suivante[p]]
        if(rang[f][p] < rang[f][fiancé[f]])
        {
            q=fiancé[f]
            fiancé[f]=p
            p=q
        }
    }
}
```

Au pire, chaque homme parcourt l'ensemble de sa liste de préférences, la complexité maximale de cet algorithme est donc linéaire en la taille de l'ensemble de ces listes.

Septième partie

La programmation dynamique

20 Retour aux plus courts chemins

En étudiant les plus courts chemins, nous avons observé que dans un digraphe acyclique, le tri topologique permet de ne traiter un sommet que si tous ses prédécesseurs sont traités.

L'idée de l'algorithme de Bellman, que nous rappelons ci-dessous, est qu'une fois que tous les plus courts chemins entre le sommet de départ s et tous les prédécesseurs d'un sommet x sont connus, il est facile de déterminer le plus court chemin du sommet s au sommet x . Il suffit en effet, pour chaque prédécesseur y du sommet x , d'additionner la taille du plus court chemin entre les sommets s et y à la distance reprise sur l'arc entre les sommets y et x . Le résultat le plus petit indiquera le plus court chemin entre s et x .

Algorithme 2 Algorithme de Bellman

```
dist[0 ... n-1] = +infini
dist[s] = 0
renuméroter les sommets dans l'ordre topologique
for(int k=1 ; k < ordre du graphe ; ++k)
{
    j = i tel que pour tous les arcs (i,k) dans le graphe
        on ait dist[i]+mat[i][k] qui soit minimum
    dist[k] = dist[j]+mat[j][k]
    pred[k]=j
}
```

Cet algorithme peut aussi être interprété comme une application de la programmation dynamique. La programmation dynamique est une méthode permettant de résoudre un problème d'optimisation en le décomposant en étapes par le principe d'optimalité. Il s'agit d'une méthode récursive dans laquelle la solution optimale d'un sous-problème sert à résoudre le sous-problème suivant. Les différentes étapes consistent en la résolution de problèmes de plus en plus gros, jusqu'à obtenir la solution du problème de départ.

Dans le cas de l'algorithme de Bellman, le principe d'optimalité est basé sur le fait que si le plus court chemin jusqu'à un noeud k passe par un noeud j , alors le sous-chemin jusque j doit être un plus court chemin jusque j . Vu que les noeuds sont triés topologiquement, le plus court chemin de 1 à k ne peut visiter que des noeuds $j < k$. A l'étape k de l'algorithme, on connaît les plus courts chemins jusqu'aux noeuds 1 à $k-1$ et on calcule récursivement le chemin le plus court jusque k . Si la fonction $dist(k)$ représente la distance du noeud 1 au noeud k , on peut écrire cette fonction sous la forme récursive :

$$dist(k) = \min_{j=1, \dots, k-1} (dist(j) + mat(j, k))$$

avec $dist(1) = 0$. L'algorithme de Bellman fournit un algorithme "dérécursifié" pour calculer cette fonction. Considérons l'exemple suivant (où les noeuds sont déjà numérotés suivant un ordre topologique) :

Nous pouvons établir récursivement :

- $f(1) = 0$
- $f(2) = 7, f(3) = 8, f(4) = 5$
- $f(5) = 12, f(6) = 17$
- $f(7) = 21$

21 Le problème de sac-à-dos

Considérons un ensemble de n objets, chaque objet i ayant un profit p_i et un poids w_i . Le problème de sac-à-dos revient à décider quels objets mettre dans un contenant ayant une capacité donnée W de manière à maximiser le

profit total des objets choisis. Nous nous intéressons ici uniquement à la version binaire du problème dans laquelle un seul objet de chaque type est disponible.

Ce problème peut être formulé mathématiquement comme :

$$\begin{aligned} z = \max \quad & \sum_{j=1}^n p_j x_j \\ \text{s.c.} \quad & \sum_{j=1}^n w_j x_j \leq W \\ & x_j \in \{0, 1\} \quad j = 1, \dots, n \end{aligned}$$

Ce problème peut être résolu en utilisant à nouveau la programmation dynamique. Une étape de l'algorithme de programmation dynamique correspond ici à un objet. On définit $f_i(w)$ pour $i = 1, \dots, n$ et $w = 0, \dots, W$ comme étant le profit maximal pouvant être obtenu avec les objets 1 à i pour une capacité w .

Supposons que nous connaissons $f_{i-1}(w)$ — c'est-à-dire le profit maximal obtenu avec les $i - 1$ premiers objets — pour toutes les valeurs de capacité w . Observons maintenant ce qui peut être obtenu avec le i -ème objet et une capacité w donnée. Si cet objet n'est pas choisi dans la solution optimale, alors le profit maximal obtenu avec les i premiers objets est le même qu'avec les $i - 1$ premiers objets, donc $f_i(w) = f_{i-1}(w)$. Par contre, si l'objet i est choisi dans la solution optimale, il reste une capacité égale à $w - w_i$ pour les $i - 1$ premiers objets. Pour que le profit soit optimal, ces $i - 1$ premiers objets doivent maximiser le profit obtenu pour une capacité $w - w_i$. Ce profit optimal étant $f_{i-1}(w - w_i)$, et l'objet i rapportant un profit p_i , on peut donc conclure dans ce cas que $f_i(w) = p_i + f_{i-1}(w - w_i)$.

Le calcul de $f_i(w)$ étant immédiat (si $w \geq w_i$, on prend l'objet, sinon on ne peut pas le prendre), la récursion pour le problème de sac-à-dos 0/1 est dès lors donnée par :

$$f_1(w) = \begin{cases} 0 & \text{pour } w = 0, \dots, w_1 - 1, \\ p_1 & \text{pour } w = w_1, \dots, W. \end{cases}$$

$$f_i(w) = \begin{cases} f_{i-1}(w) & \text{pour } w = 0, \dots, w_i - 1, \\ \max\{f_{i-1}(w), p_i + f_{i-1}(w - w_i)\} & \text{pour } w = w_i, \dots, W. \end{cases}$$

Le profit optimal pour le problème de départ est donné par la valeur finale $f_n(W)$.

Illustrons cette récursion avec l'exemple d'un sac de capacité 19 et un ensemble de 5 objets avec les caractéristiques suivantes :

Objet	1	2	3	4	5
Profit	2	4	5	8	10
Poids	2	3	6	8	9

Les valeurs de $f_i(w)$ calculées par la récursion sont données dans le tableau ci-dessous.

	1	2	3	4	5
0	0	0	0	0	0
1	0	0	0	0	0
2	2	2	2	2	2
3	2	4	4	4	4
4	2	4	4	4	4
5	2	6	6	6	6
6	2	6	6	6	6
7	2	6	6	6	6
8	2	6	7	8	8
9	2	6	9	9	10
10	2	6	9	10	10
11	2	6	11	12	12
12	2	6	11	12	14
13	2	6	11	14	14
14	2	6	11	14	16
15	2	6	11	14	16
16	2	6	11	15	16
17	2	6	11	17	18
18	2	6	11	17	19
19	2	6	11	19	20

Le profit maximal est donc égal à 20. La récursion nous donne uniquement la valeur du profit optimal. Pour retrouver les objets présents dans la solution, on peut effectuer une récursion en arrière.

En effet, observons dans le tableau que $f_5(20) \neq f_4(20)$, ce qui signifie que l'objet 5 fait partie de la solution optimale. Dès lors, la combinaison des 4 premiers objets est celle qui donne un profit égal à $f_4(19 - 9 = 10)$. Comme $f_4(10) \neq f_3(10)$, l'objet 4 fait également partie de la solution optimale. La combinaison des 3 premiers objets est celle qui donne un profit égal à $f_3(10 - 8 = 2)$. On observe alors que $f_3(2) = f_2(2) = f_1(2)$, ce qui signifie que les objets 3 et 2 ne sont pas sélectionnés alors que l'objet 1 l'est. La solution optimale est donc composée des objets 1, 4 et 5.

On peut observer énormément de redondance dans le tableau. De plus, la taille du tableau peut "exploser" très vite si la valeur numérique de la capacité est grande. Or, nous ne sommes intéressés que par les combinaisons de profit et capacité où la capacité utilisée est minimale pour un profit donné. A partir de cette observation, il est intéressant de définir la notion de solution dominée : une solution est dominée par une autre si elle utilise plus de capacité pour un profit inférieur ou égal.

Au lieu de calculer les valeurs de $f_i(w)$ pour toutes les valeurs de w à l'itération i , on peut se contenter de construire uniquement les paires (profit, poids) non dominées. A chaque étape, on ajoute un objet. Les paires (profit, poids) considérées sont celles de l'étape précédente, plus celles obtenues par ajout de l'objet courant aux paires de l'étape précédente. Toutes les paires dominées sont éliminées au fur et à mesure de leur détection.

Pour l'exemple précédent, nous obtenons :

Itération 0 : (0,0)

Itération 1 : (0,0), (2,2)

Itération 2 : (0,0), (2,2), (4,3), (6,5)

Itération 3 : (0,0), (2,2), (4,3), ~~(5,6)~~, (6,5), (7,8), (9,9), (11,11)

Itération 4 : (0,0), (2,2), (4,3), (6,5), ~~(7,8)~~, (8,8), (9,9), (10,10), ~~(11,11)~~, (12,11), (14,13), (15, 16), (17,17), (19,19)

Itération 5 : (0,0), (2,2), (4,3), (6,5), (8,8), ~~(9,9)~~, (10,9), ~~(10,10)~~, (12,11), (14,13), ~~(15,16)~~, (16,14), ~~(17,17)~~, (18,17), (19,18), ~~(19,19)~~, (20,19)

où les paires soulignées représentent les paires ajoutées, et les paires dominées sont barrées. L'écriture d'un algorithme détaillé pour le calcul des paires non dominées est laissé en guise d'exercice.

22 Détermination de la taille des lots

Le problème de la détermination de la taille des lots (lot sizing en anglais) est un problème arrivant dans le contexte de la gestion de production. Etant donnés des besoins nets pour un produit, répartis dans le temps, le problème

est de décider quand produire et en quelles quantités pour satisfaire ces besoins nets de façon la plus économique possible.

La solution de ce problème représente un compromis entre le coût de lancement l encouru au début de chaque période pendant laquelle une production a lieu, et le coût de stockage s par unité stockée et par période, qui intervient lorsque la production pour satisfaire la demande de plusieurs périodes est regroupée afin de diminuer les coûts de lancement.

Nous nous intéressons ici au cas particulier où les coûts de lancement et de stockage sont constants (ne dépendent pas de la période). Dans ce cas, la programmation dynamique peut être utilisée pour obtenir une solution optimale du problème.

Soit d_t la demande pour la période t . En introduisant les variables

$$\begin{aligned} x_t & \text{ quantité produite à la période } t, \\ i_t & \text{ stock à la période } t, \\ y_t & \text{ indicateur de production à la période } t \text{ (0/1),} \end{aligned}$$

le problème peut être formulé mathématiquement comme

$$\begin{aligned} \min \quad & l \sum_{t=1}^T y_t + s \sum_{t=1}^T i_t \\ \text{s.c.} \quad & i_t = i_{t-1} + x_t - d_t \quad t = 1, \dots, T \\ & x_t \leq M y_t \quad t = 1, \dots, T \\ & i_0 = i_T = 0 \\ & x_t, i_t \geq 0, y_t \in \{0, 1\} \quad t = 1, \dots, T \end{aligned}$$

Remarquons que si les périodes de productions (i.e. les valeurs des variables y_t) sont fixées, ce problème se ramène à un problème de flot de coût minimum dans un réseau particulier.

De plus, on peut montrer que les propriétés suivantes sont satisfaites :

1. Il existe une solution optimale telle que $i_{t-1}x_t = 0$ pour tout t (production uniquement quand stock vide).
2. Il existe une solution optimale telle que si $x_t > 0$, $x_t = \sum_{i=t}^{t+k} d_i$ pour un $k > 0$.

En d'autres termes, il existe toujours une solution optimale telle que la production a lieu uniquement lorsque le stock est vide, et pour satisfaire la demande d'un nombre entier de périodes consécutives. Cette observation nous permet de construire un algorithme de programmation dynamique. Soit $H(k)$ le coût minimum d'une solution couvrant la demande des périodes 1 à k .

Si $t \leq k$ est la dernière période de production entre 1 et k , alors le coût de production pour les périodes 1 à $t-1$ doit être optimal et égal à $H(t-1)$.

On en déduit la récursion suivante :

$$H(k) = \min_{1 \leq t \leq k} \{H(t-1) + l + s \sum_{i=t+1}^k (i-t)d_i\}$$

avec la condition initiale $H(0) = 0$.

Considérons un exemple avec 6 périodes et les demandes nettes suivantes :

Période	1	2	3	4	5	6
Besoins nets	80	100	125	100	50	50

Supposons que le coût de lancement est $l = 300$, et le coût de stockage $s = 2.5$ par unité et par période.

En appliquant la récursion, nous obtenons :

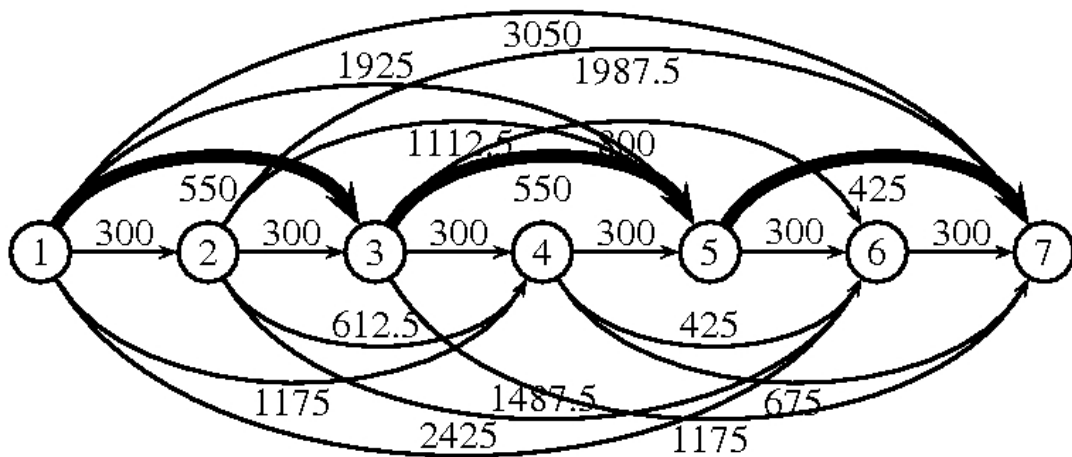
$$\begin{aligned}
 H(0) &= 0 \\
 H(1) &= \underline{0 + 300} = 300 \\
 H(2) &= \min\{\underline{0 + 300 + 2.5 * 100}, 300 + 300\} = 550 \\
 H(3) &= \min\{0 + 300 + 2.5 * (2 * 100 + 125), 300 + 300 + 2.5 * 125, \underline{550 + 300}\} = 850 \\
 H(4) &= \min\{0 + 300 + 2.5 * (3 * 100 + 2 * 125 + 100), 300 + 300 + 2.5 * (2 * 125 + 100), \\
 &\quad \underline{550 + 300 + 2.5 * 100}, 850 + 300\} = 1100 \\
 H(5) &= \min\{0 + 300 + 2.5 * (4 * 100 + 3 * 125 + 2 * 100 + 50), 300 + 300 + 2.5 * (3 * 125 + 2 * 100 + 50), \\
 &\quad 550 + 300 + 2.5 * (2 * 100 + 50), \underline{850 + 300 + 2.5 * 50}, 1100 + 300\} = 1275 \\
 H(6) &= \min\{0 + 300 + 2.5 * (5 * 100 + 4 * 125 + 3 * 100 + 2 * 50 + 50), \\
 &\quad 300 + 300 + 2.5 * (4 * 125 + 3 * 100 + 2 * 50 + 50), 550 + 300 + 2.5 * (3 * 100 + 2 * 50 + 50), \\
 &\quad 850 + 300 + 2.5 * (2 * 50 + 50), \underline{1100 + 300 + 2.5 * 50}, 1275 + 300\} = 1525
 \end{aligned}$$

La solution optimale a donc un coût de 1525. Les valeurs soulignées correspondent au choix pour lequel le minimum est atteint à chaque étape. Pour retrouver les périodes de production, on peut soit appliquer une récursion en arrière, soit retenir à chaque étape la valeur de la période de production précédente t pour lequel le minimum est atteint. Dans ce cas-ci, on en déduit que la production à lieu aux périodes 1, 3 et 5, avec les quantités produites reprises dans le tableau ci-dessous.

Période	1	2	3	4	5	6
Besoins nets	80	100	125	100	50	50
Production	180		225		100	

Comme tout algorithme de programmation dynamique, la fonction H peut aussi être interprétée comme la calcul de plus courts chemins dans un graphe où les noeuds représentent les périodes de production et un arc de t à k représente le fait de produire en période t pour satisfaire les demandes de la période t à la période $k - 1$, avec un coût égal à $l + s \sum_{i=t+1}^k (i - t)d_i$.

Dans notre exemple, le problème se ramène à trouver un plus court chemin du noeud A au noeud F dans le graphe suivant :



Le chemin optimal est indiqué en gras.