

Optimisation et Recherche Operationnelle

Zakaria EJJED

Contents

Thursday January 26 2023 Course	2
I - Problème d'ordonnancement	4
Quelques formules:	7
Thursday January 26 2023 Exercises	7
Exercice 1	7
Exercice 2	8
Thursday February 2nd 2023 Course	11
II - Programmation linéaire	15
Thursday February 2nd 2023 Exercises	18
Exercice 1	18
Exercice 2 (annule et remplace !)	18
Thursday February 9th 2023	19
Exercice 3	19
Exercice 4	19
Exercice 5	21
Algo du simplexe	22
III - Séparation et Evaluation	22

Thursday January 26 2023 Course

Optimisation:

→ on cherche le point x^* où l'optimisation (minimum ou maximum selon le cas) de la fonction f est réalisé.

Recherche opérationnelle: → on se trouve dans un contexte industriel dans lequel on cherche à résoudre un problème concret, problèmes que l'on peut représenter comme des problèmes d'optimisation.

⇒ modélisation: travail de représentation mathématique/informatique d'un problème concret qui est posé. → abstraction/simplification.

Exemple: Gestion de stocks.

- **Entrepôt:** Chaque jour, $q=100$ unités sortant; une unité en stock coûte $a=0,16\text{€/jour}$; le coût d'une passation de commande est $b=50\text{€}$, et le réapprovisionnement à lui automatiquement dès que le stock est vide; une commande ne peut excéder $c=400$ unités.
On cherche le volume optimal O^R d'une commande.
→ modélisation: on cherche à minimiser le coût de gestion de l'entrepôt.
- **Coût de gestion:**

$$\begin{array}{rcl} \text{coût de stock} & = & 0,16\text{€/jour/unité} \\ + \text{ coût de commande} & = & 50\text{€} \end{array}$$

→ On exprime tout en euros par jour.

- **Coût de stock:**
On cherche le nombre moyen d'unités dans le stock: on suppose que le stock se vide à un rythme régulier, et de façon continue.
//1 Le stock moyen est le stock moyen sur une période $\frac{Q}{2}$.

$$\int_0^T Q(1 - \frac{t}{T})dT = \frac{Q}{2}$$

Le coût quotidien de gestion de stock est $a \cdot \frac{Q}{2} \text{€/j.}$

coût de commande:

$\frac{Q}{100}$ est le temps (en jour) entre deux commandes: on fait une commande, coûte $b=50\text{€}$, tous les $\frac{Q}{100}$ jours, et le coût quotidien moyen de commande est des:

$$\frac{b}{a/q} = \frac{50}{Q/100} = \frac{5000}{Q}$$

Le coût quotidien moyen de gestion de l'entrepôt est de:

$$\begin{aligned}
f(Q) &= 0,08Q + \frac{5000}{Q} \\
D &= [0, 400] \quad Q \leq 400 \quad Q \geq 0 \\
\min \{f(Q)/Q \in D\} \\
\min \{0,08Q + \frac{5000}{Q}/0 \leq 400\}
\end{aligned}$$

On détermine le table de variation de f:

$$\begin{aligned}
\forall Q \in]0, 400], \\
f'(Q) &= 0,08 - \frac{5000}{Q^2} \\
f'(Q) &= 0 \\
\Leftrightarrow 0,08 - \frac{5000}{Q^2} &= 0 \\
\Leftrightarrow \frac{5000}{Q^2} &= 0,08 \\
\Leftrightarrow \frac{Q^2}{5000} &= \frac{1}{0,08} \\
\Leftrightarrow Q^2 &= \frac{5000}{0,08} = \frac{500000}{8} = 62500 \\
\Leftrightarrow Q &= \sqrt{62500} \simeq 250 \text{ car } Q > 0
\end{aligned}$$

$$\text{si } \sqrt{\frac{2bq}{Q}}$$

...

//TABLEAU DE VARIATION

La minimisation de f est atteinte à 250: le volume optimal de commande est Q=250, pour un coût quotidien de f(Q)=40.

Si on généralise à des a, b, c, q, quelconques:

$$\begin{aligned}
\forall Q \in]0, 0], \\
f(Q) &= a \times \frac{Q}{Q^2} + \frac{bq}{Q} \\
f'(Q) &= \frac{a}{2} - \frac{bq}{Q^2} \\
f'(Q) &= 0 \\
\Leftrightarrow \frac{a}{2} - \frac{bq}{Q^2} &= 0 \\
\Leftrightarrow \frac{bq}{Q^2} &= \frac{a}{2} \\
\Leftrightarrow \frac{Q^2}{5000} &= \frac{1}{0,08} \\
\Leftrightarrow Q &= \sqrt{\frac{2bq}{a}} \text{ car } Q > 0
\end{aligned}$$

//TABLEAU DE VARIATIONS

Algorithm 1 Algo Stock opti (a,b,c,q:réels > 0)→Q*:réel>0

```

if  $\sqrt{\frac{2bq}{a}} \leq c$  then
     $Q^* \leftarrow \frac{2bq}{a}$ 
else
     $Q^* \leftarrow c$ 
end if

```

I - Problème d'ordonnancement

On cherche à déterminer l'ordre dans le quel effectuer des tâches, définies par leurs durées, et par des relations de précédence: certaines tâches ne peuvent être effectuées qu'après que d'autres aient été accomplies.

On cherche à déterminer le planning permettant de terminer au plus vite le projet.

On a une série de tâche $\{t_i/i = 1, \dots, u\}$, A chaque tâche est associée une durée, ainsi qu'une liste de tâches $\{p_{ij}/j = 1, \dots, k_i\}$ qui doivent être terminée avant que la tâche t_i ne puisse commencer.

La fonction objective est la durée du projet, que l'on cherche à minimiser; les contraintes sont les contraintes de précédence; les variables de décision décrivent l'ordre dans lequel on effectue les tâches: c'est le planning de ces tâches, c'est à dire la date de début de chacune des tâches.

La durée totale du projet, fonction objective, est la durée entre le début de la première tâche ($\rightarrow t = 0$) et la fin de la dernière.

$t_f =$

la tâche t_i se termine au temps a_i (date de début de la tâche)+ d_i (durée de la tâche)

Le projet se termine en même temps que la dernière tâche, au temps:
 $\max\{ a_i + d_i/i = 1, \dots, u \}$.

On cherche donc à minimiser $\max\{ a_i + d_i/i = 1, \dots, u \}$.

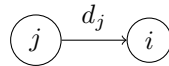
On a de plus les contraintes suivantes:

$$\begin{aligned}
 \forall i = 1, \dots, u \\
 a_i &\geq 0 \\
 \forall i = 1, \dots, u \quad \forall k \in P_i, \\
 a_k + d_k a &\leq a a_i
 \end{aligned}$$

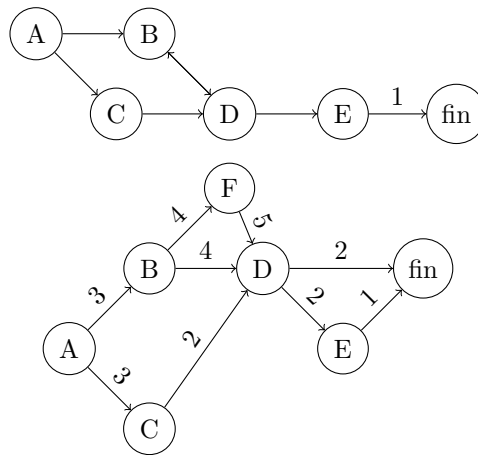
Au final, le problème consiste à trouver des a_i réalisant:

$\min\{\max\{a_i + d_i/i = 1, \dots, u\} / \forall i, a_i \geq 0; \quad \forall k \in P_i, a_i \geq a_k + d_k\}$ contraintes

On peut également modéliser ce problème sous la forme de graphes. Dans la méthode des **potentiels**, chaque tâche est représentée par un sommet et les relations de précédences par des arcs. Un arc joint j à i si $j \in P_i$



A chaque arc, on associe la durée de la tâche dont il est issu. La longueur d'un chemin représente la durée de tâches qui doivent être exécutés consécutivement: la durée du projet est donc plus grande que la longueur et n'importe quel chemin dans ce graphe.



Recherche d'un plus long chemin dans un graphe (problème de maximisation):
 $\max\{f\} = -\min\{-f\}$

//COURBE

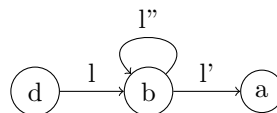
L'algo de Dijkstra ne peut s'appliquer car il nécessite des valuations positives.
 \Rightarrow algo de **Bellman-Ford**.

La durée totale du projet est donc minorée par la longueur du plus long chemin dans ce graphe; en fait, on peut la rendre égale à cette longueur.

\rightarrow recherche d'un plus long chemin dans un graphe.

\Leftrightarrow revient à chercher le plus court chemin dans ce graphe dans lequel on a remplacé chaque valuation par son opposé.

Le problème du plus court chemin n'a pas de solution en présence d'un cycle de poids <0 ;



$$l + l' > l + l' + l'' > l + l' + 2l'' > \dots > l + l' + kl'' \quad \forall k$$

→ pas de plus court chemin, parce qu'une fois qu'on en a un, on peut toujours en construire un encore plus court.

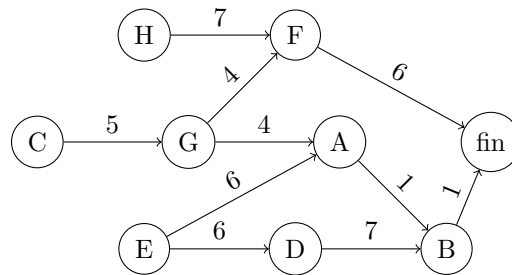
DAG: directed acyclic graph, graphe orienté acyclique.

⇒ algo de **Ford**

On prend les noeuds dans un ordre garantissant qu'un noeud est traité après tous ses prédécesseurs (cet ordre existe sinon le graphe contiendrait un cycle)
traiter le noeud i consiste à fixer sa date à $\max\{a_j + d_j / j \in P\}$

Par exemple:

Tâche	Description	Durée	Précédence
A	teinture	1	E,G
B	confection	1	A,D
C	étude de marché	5	-
D	commande fermeture éclair	7	E
E	commande tissu	6	-
F	impression catalogue	6	G,H
G	choix coloris	4	C
H	contact imprimeur	7	-



Graphe des potentiels

$$\max\{a_j + d_j / j \in P\}$$



Tâche	Date au plus tôt	Date au plus tard	Marge totale	Marge libre
C	0	$\min\{5-5\}=0$	0	0
E	0	$\min\{13-6,7-6\}=1$	1	0
H	0	$\min\{5-7\}=2$	2	2
D	$\max\{0+6\}=6$	$\min\{14-7\}=7$	1	0
G	$\max\{0+5\}=5$	$\min\{13-4,9-4\}=5$	0	0

Tâche	Date au plus tôt	Date au plus tard	Marge totale	Marge libre
A	$\max\{5+4,0+6\}=9$	$\min\{14-1\}=13$	4	3
F	$\max\{5+4,0+7\}=9$	$\min\{15-6\}=9$	0	0
B	$\max\{9+1,6+7\}=13$	$\min\{15-1\}=14$	1	1
fin	$\max\{5+6,13+2\}=15$	$=15$	0	0

↓

Planning optimal

Les tâches C,G et F doivent être exécutées consécutivement, et elles prennent à elles 3 15j: le projet ne peut pas durer moins, et comme le planning trouvé prend 15j, il est optimal.

On appelle date au plus tard la date avant laquelle une tâche doit commencer au risque de mettre tout le projet en retard.

Une tâche ayant une marge totale de 0 est dite critique: il existe un chemin partant d'un sommet sans prédécesseur et allant au noeud "**fin**" en ne passant que par des tâches critiques.

La marge libre est le retard que peut prendre une tâche tout en permettant aux tâches suivantes de commencer à leur date au plus tôt (=conserver leur marge).

Quelques formules:

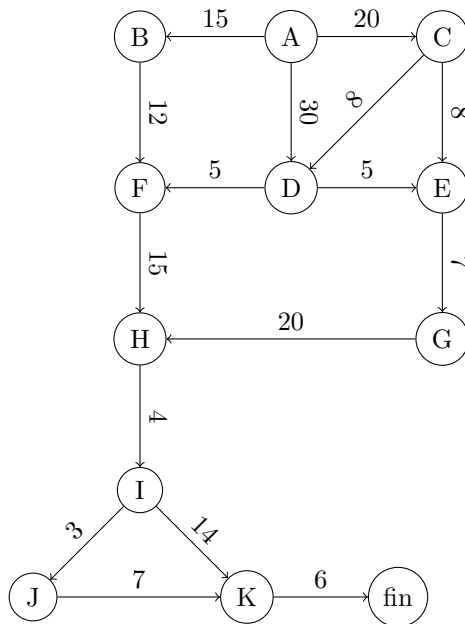
$$\begin{aligned}
 \text{tôt}_i &= \max\{\text{tôt}_j + d_j / j \in P_i\} \\
 &\quad (\text{avec } \max \emptyset = 0) \\
 \text{tard}_i &= \min\{\text{tard}_j - d_i / i \in P_j\} \\
 \text{marge}_i &= \text{tard}_i - \text{tôt}_i \\
 \text{libre}_i &= \min\{\text{tôt}_j - d_i - \text{tôt}_i / i \in P_j\}
 \end{aligned}$$

Thursday January 26 2023 Exercises

Exercice 1

Tâche	Durée	Antériorité
A	30	-
B	12	A+15j
C	8	A+20j
D	5	A et C
E	7	C et D
F	15	B et D
G	20	E
H	4	F et G
I	14	H

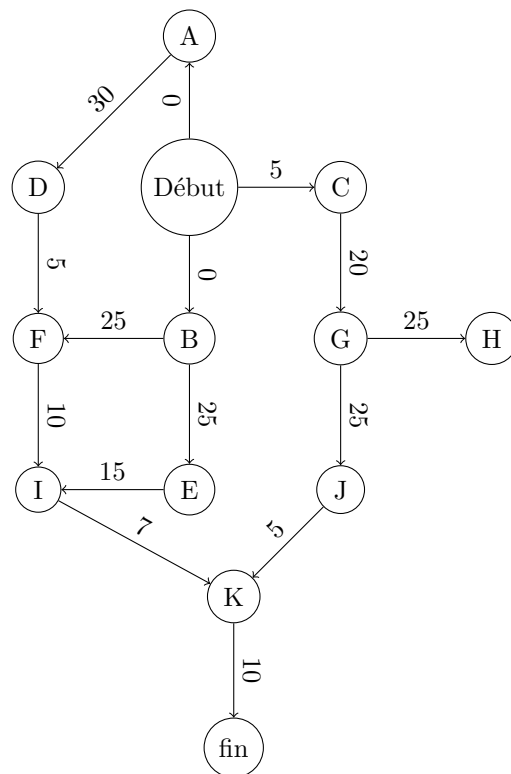
Tâche	Durée	Antériorité
J	7	I+3j
K	6	I et J



Tâche	Date au plus tôt	Date au plus tard	Marge totale	Marge libre
A	0	0	0	0
B	15	35	20	8
C	20	22	2	2
D	30	30	0	0
E	35	35	0	0
F	35	47	12	12
G	42	42	0	0
H	62	62	0	0
I	66	66	0	0
J	69	73	4	0
K	80	80	0	0
fin	86	86	0	0

Exercice 2

Tâche	Durée	Antériorité
A	30	-
B	25	-
C	20	5j après début
D	5	A
E	15	B
F	10	B,D
G	25	C
H	12	G
I	7	E,F
J	5	G
K	10	I,J



Tâche	Date au plus tôt	Date au plus tard	Marge totale	Marge libre
Début	0	0	0	0
A	0	0	0	0
B	0	0	0	0
C	5	5	0	0
D	30	33	3	0

Tâche	Date au plus tôt	Date au plus tard	Marge totale	Marge libre
E	25	33	8	5
F	35	38	3	0
G	25	25	0	0
H	50	50	0	0
I	45	48	3	3
J	50	50	0	0
K	55	55	0	0
fin	65	65	0	0

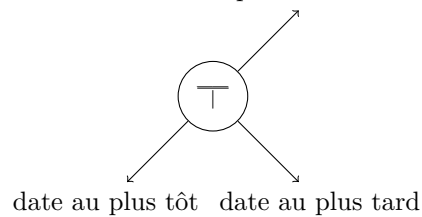
Thursday February 2nd 2023 Course

La méthode ***PERT*** est une autre méthode pour calculer un ordonnancement optimal. Dans cette méthode, on dessine un graphe dont les arcs (et non plus les noeuds) représentent les tâches du problème d'ordonnancement.

Les noeuds représentent alors des “étapes” de la réalisation du projet, étapes auxquelles un certain nombre de tâches ont été réalisées, permettant de démarrer de nouvelles tâches.

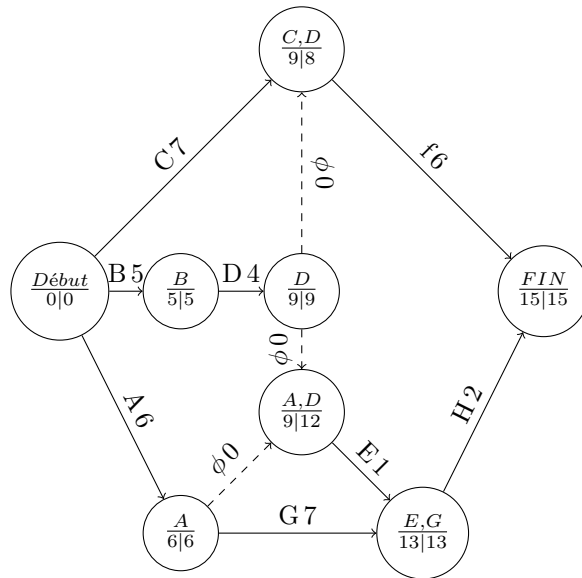
Dans des situations où deux tâches n'ont pas les mêmes antécédents, mais en partagent certains, pour placer leur étape de départ, on peut être obligé de recourir à des tâches fictives, de durées nulles, et allant d'une étape correspondant à la réalisation d'un ensemble de ses tâches vers une étape correspondant à un ensemble de tâches.

ensemble de tâches terminées correspondant à l'antécedence des tâches qui en sortent.



1.

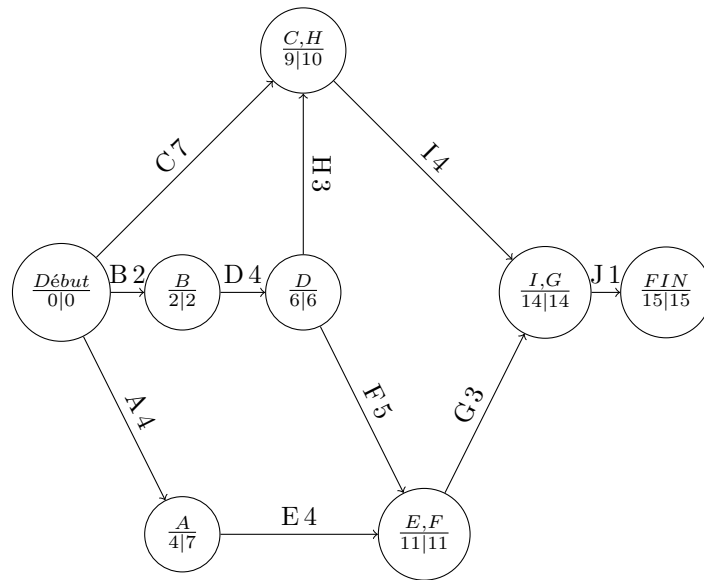
Tâche	durée	antécédant
A	6	-
B	5	-
C	7	-
D	4	B
E	1	A,D
F	6	C,D
G	7	A
H	2	E,G



Tâche	date au plus tôt	date au plus tard	marge totale	marge libre
A	0	0	0	0
B	0	0	0	0
C	0	2	2	2
D	5	5	0	0
E	9	9	0	0
F	9	9	0	0
G	6	6	0	0
H	13	13	0	0

2.

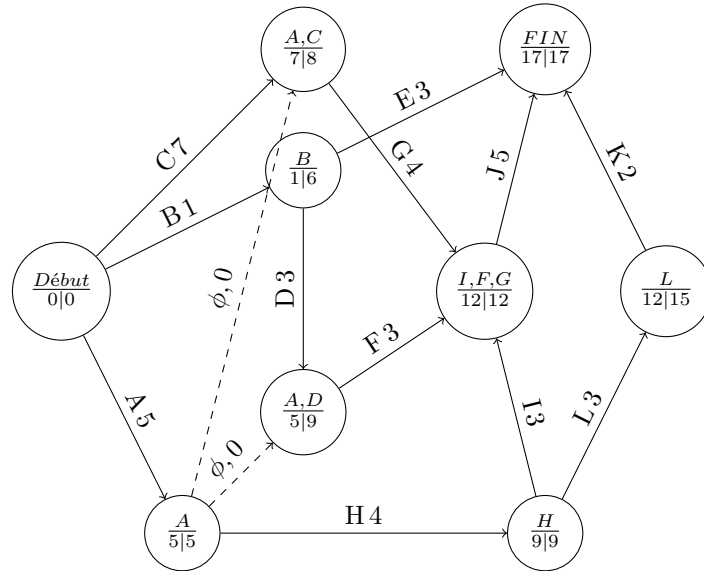
Tâche	durée	antécédant
A	4	-
B	2	-
C	7	-
D	4	B
E	4	A
F	5	D
G	3	E,F
H	3	D
I	4	C,H
J	1	I,G



Tâche	date au plus tôt	date au plus tard	marge totale	marge libre
A	0	3	3	0
B	0	0	0	0
C	0	3	3	2
D	2	2	0	0
E	4	7	3	3
F	6	6	0	0
G	11	11	0	0
H	6	7	1	0
I	9	10	1	1
J	14	14	0	0

3.

Tâche	durée	antécédant
A	5	-
B	1	-
C	7	-
D	3	B
E	3	B
F	3	A,D
G	4	A,C
H	4	A
I	3	H
J	5	I,F,G
K	2	L
L	3	H



Tâche	date au plus tôt	date au plus tard	marge totale	marge libre
A	0	0	0	0
B	0	5	5	0
C	0	1	1	0
D	1	6	5	1
E	1	14	13	13
F	5	9	4	4
G	7	8	1	1
H	5	5	0	0
I	9	9	0	0
J	12	12	0	0

Tâche	date au plus tôt	date au plus tard	marge totale	marge libre
K	12	15	3	3
L	9	12	3	0

II - Programmation linéaire

Un programme linéaire est un problème d'optimisation dans lequel la fonction objective est linéaire et les contraintes sont affines.

$\max/\min \{ \sum_{i=1}^n c_i x_i / \forall j, \sum_{j=1}^n a_{ij} x_i \{ \leq, \geq, \text{ ou } = \} b_j \}$

C'est une catégorie particulière, mais importante de problèmes d'optimisation.

- Imaginer une entreprise fabriquant des compotes.
→ compote pommes/fraises PF
→ compote pommes P
→ compote pommes F
- Pour produire 1kg de compotes
PF → 2kg pommes, 1kg fraises
P → 3kg pommes
F → 1kg pommes, 2kg fraises

Chaque jour, l'usine reçoit, 3t de pommes et 5t de fraises.

- La préparation d'une tonne de compote nécessite:
→ 40h pour PF
→ 30h pour F
→ 40h pour P
L'entreprise dispose de 80h de travail/j.

L'entreprise ne peut pas produire plus de 4t de compote par jour (limite de stockage est expédition).

- Le bénéfice de l'entreprise est de:
→ 60€ par tonne de PF
→ 70€ par tonne de F
→ 20€ par tonne de P
- Les variables de décision sont ici les nombres de tonne de chaque compote produits chaque jour:
→ x_1 est le nombre de tonnes de PF produit chaque jour.
→ x_2 est le nombre de tonnes de F produit chaque jour.
→ x_3 est le nombre de tonnes de P produit chaque jour.
 (x_1, x_2, x_3) représente un plan de production.
- Ce plan de production doit vérifier des contraintes:
 - stockage:
 $x_1 + x_2 + x_3 \leq 4$

- travail:
 $40x_1 + 30x_2 + 40x_3 \leq 80$
- approvisionnement:
 $2x_1 + x_2 + 3x_3 \leq 3$
 \uparrow
quantité de pommes (en t)
nécessaire pour produire
la quantité de PF du plan
de produit.

$$x_1 + 2x_2 \leq 5$$

- positivité:
 $x_1 \geq 0$
 $x_2 \geq 0$
 $x_3 \geq 0$

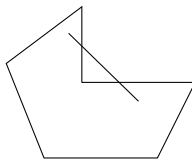
On cherche à maximiser les bénéfices quotidiens:

$$60x_1 + 70x_2 + 20x_3$$

Au final, le problème peut s'écrire:

$$\max\{60x_1 + 70x_2 + 20x_3\} \text{ s.c } \begin{cases} x_1 + x_2 + x_3 & \leq 4 \\ 40x_1 + 30x_2 + 40x_3 & \leq 80 \\ 2x_1 + x_2 + 3x_3 & \leq 3 \\ x_1 + 2x_2 & \leq 5 \\ x_1, x_2, x_3 & \leq 0 \end{cases}$$

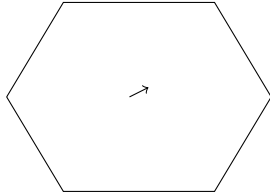
On appelle solution (réalisable) d'un problème d'optimisation un plan de production x vérifiant les contraintes. La solution optimale x^* est la solution réalisable optimisant la fonction objective.



pas convexe

- L'ensemble des contraintes définit polyèdre:
 \rightarrow c'est l'intersection des sous-espaces définis par chaque contrainte,
 \rightarrow chaque contrainte $\sum a_{ij}x_i \leq b_j$ définit un demi-espace limité par l'hyperplan $\sum a_{ij}x_i = b_j$
 La fonction objective est linéaire. Donc son gradient est constant.
 Donc son gradient est constant. De plus les lignes de niveau associée à

cette fonction sont des hyperplans.



Si on prend un point à l'intérieur du polyèdre, le gradient en ce point est non-nul, et on peut prendre un point $x_0 + \epsilon \nabla f(x_0)$ avec $\epsilon > 0$ suffisamment petit, point qui sera meilleur que x_0 et toujours dans le polyèdre.

La solution optimale est forcément sur la frontière ("au bord") de polyèdre.

//COURBE ORO 1

Une solution optimale est donc nécessairement un sommet du polyèdre. Il suffit donc de parcourir les sommets du polyèdre. De plus le polyèdre est convexe.



La convexité garantit que si un sommet a une valeur meilleur que celle des sommets adjacents, il est un optimum global.

On peut donc parcourir les sommets de façon à examiner des sommets avec une valeur toujours meilleure, jusqu'à trouver l'optimum.

gradient:

$$\nabla f(y) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(y) \\ \frac{\partial f}{\partial x_2}(y) \\ \frac{\partial f}{\partial x_3}(y) \end{pmatrix}$$

courbe de niveau:

$$\{x : f(x) = \alpha\}$$

En tout point x_0 , le gradient $\vec{\nabla} f(x_0)$ est normale à la courbe de niveau

$$\{x / f(x) = f(x_0)\}$$

Thursday February 2nd 2023 Exercises

Exercice 1

$$\max \{ 15x_1 + 8x_2 - 6HS - 1,5MP - PUB_1 - PUB_2 \}$$

$$\text{sc} \begin{cases} MP & \leq 400 \\ 2x_1 + x_2 & \leq MP \\ 0,75x_1 + 0,5x_2 & \leq 160 + HS \\ x_1 & \leq 50 + 10PUB_1 \\ x_2 & \leq 60 + 15PUB_2 \\ PUB_1 + PUB_2 & \leq 100 \\ x_1, x_2, MP, HS, PUB_1, PUB_2 & \geq 0 \end{cases}$$

Exercice 2 (annule et remplace !)

On considère un problème d'ordonnancement,

tâche	durée	antécédents
i	d_i	P_i

Modéliser le sous forme de PL.

Notons x_i la date de début de la tâche i (en u.t.) contraintes:

$$\begin{aligned} \forall i, \forall j \in P_i, x_i &\geq x_j + d_j \\ \forall i, x_i &\geq 0 \end{aligned}$$

fonction objective:

$$\min\{\max\{x_i + d_i\}\}$$

Cette fonction objective n'est pas linéaire.

On ajoute une variable de décision, x_f , représentant la date de fin du projet.

$$\begin{aligned} &\forall, x_f \geq x_i + d_i \\ &\min\{x_f\} \\ \text{sc} \begin{cases} \forall i, \forall j \in P_i, x_i \geq x_j + d_j \\ \forall i, x_i \geq 0 \\ \forall i, x_f \geq x_i + d_i \end{cases} \end{aligned}$$

x_f représente une date plus grande que le $\max\{x_i + d_i\}$, mais que les contraintes autorisent à lui être égale; puisqu'on cherche à le minimiser, on a la garanti que, dans la solution optimale, il vaudra ce max.

Thursday February 9th 2023

Exercice 3

Notons x_1 la quantité de brut n°1, en Mt, traité par la raffinerie.

Notons x_2 la quantité de brut n°2, en Mt, traité par la raffinerie.

Notons x_3 la quantité de brut n°3, en Mt, traité par la raffinerie.

On cherche à minimiser le bénéfice:

$$4x_1 + 5x_2 + 5x_3$$

A cause de la capacité d'entreposage et de livraison des gaz et gaz liquéfiés, le plan de production (x_1, x_2, x_3) doit respecter:

$$0,02x_1 + 0,06x_3 \leq 0,3 \Leftrightarrow 2x_1 + 6x_3 \leq 30$$

- Pour l'essence:

$$20x_1 + 25x_2 + 30x_3 \leq 105 \Leftrightarrow 4x_1 + 5x_2 + 6x_3 \leq 21$$

- Pour le pétrole:

$$8x_1 + 4x_3 \leq 18$$

- Pour le gasoil:

$$40x_1 + 25x_2 + 30x_3 \leq 135$$

Exercice 4

Notons x_1 la fraction d'orge contenu dans l'aliment produit, en t. Notons x_2 la fraction d'arachide contenu dans l'aliment produit, en t. Notons x_3 la fraction de sésame contenu dans l'aliment produit, en t.

On cherche à minimiser le coût d'une tonne d'aliment produit:

$$25x_1 + 41x_2 + 39x_3$$

Un aliment de composition (x_1, x_2, x_3) contient $0,12x_1 + 0,52x_2 + 0,42x_3$, qui doit être **au moins** égale à 0,22:

$$12x_1 + 52x_2 + 42x_3^* \geq 22$$

*comme taux de protéines

- Pour la graisse, on a:

$$2x_1 + 2x_2 + 10x_3 \geq 3,6$$

De plus:

$$\begin{array}{rcl} x_1, x_2, x_3 & \geq & 0 \\ x_1 + x_2 + x_3 & = & 1 \end{array}$$

- Au final:

$$\min\{25x_1 + 41x_2 + 39x_3\}$$

$$\text{sc} \begin{cases} 12x_1 + 52x_2 + 42x_3 & \geq 22 \\ 2x_1 + 2x_2 + 10x_3 & \geq 3,6 \\ x_1 + x_2 + x_3 & = 1 \\ x_1, x_2, x_3 & \geq 0 \end{cases}$$

Puisque $x_1 + x_2 + x_3 = 1$, on substitue à x_3 $1 - x_1 - x_2$ partout. Le plan de production, de la composition de l'aliment, est $(x_1, x_2, 1 - x_1 - x_2)$.

- La première contrainte devient:

$$\begin{array}{rcl} 12x_1 + 52x_2 + 42(1 - x_1 - x_2) & \geq & 22 \\ \Leftrightarrow -30x_1 + 10x_2 + 42 & \geq & 22 \\ \Leftrightarrow -30x_1 + 10x_2 & \geq & -20 \\ \Leftrightarrow 3x_1 - x_2 & \geq & 2 \end{array}$$

- La deuxième contrainte devient:

$$\begin{array}{rcl} 8x_1 + 8x_2 & \leq & 6,4 \\ \Leftrightarrow x_1 + x_2 & \leq & 0,8 \\ \Leftrightarrow 5x_1 + 5x_2 & \leq & 4 \end{array}$$

- La troisième contrainte devient:

$$0 = 0$$

On a de plus:

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

$$\begin{array}{rcl} 1 - x_1 - x_2 & \geq & 0 \\ \Leftrightarrow x_1 + x_2 & \leq & 1 \end{array}$$

- La fonction objective devient:

$$\begin{aligned} & \min\{25x_1 + 41x_2 + 39(1 - x_1 - x_2)\} \\ = & \min\{-14x_1 + 2x_2 + 39\} \\ = & \min\{-14x_1 + 2x_2\} + 39 \end{aligned}$$

- On cherche:

$$\min\{-14x_1 + 2x_2\} + 39$$

$$\begin{cases} 3x_1 - 2x_2 & \leq 2 \\ 5x_1 + 5x_2 & \leq 4 \\ x_1 + x_2 & \leq 1 \\ x_1, x_2 & \geq 0 \end{cases}$$

Cette contrainte est redondante avec la précédente: $4 \leq 5$, donc

$$5x_1 + 5x_2 \leq 4 \Rightarrow 5x_1 + 5x_2 \leq 5 \Rightarrow x_1 + x_2 \leq 1$$

Exercice 5

Notons x_1 le nombre d'objets A_1 produit chaque mois. Notons x_2 le nombre d'objets A_2 produit chaque mois. Notons x_3 le nombre d'objets A_3 produit chaque mois.

- On cherche à maximiser le bénéfice mensuel:

$$60x_1 + 40x_2 + 80x_3$$

- On a:

$$\begin{aligned} x_1 & \leq 4800 \\ x_2 & \leq 5400 \\ x_3 & \leq 2000 \end{aligned}$$

à cause des contraintes du marché.

- La disponibilité de la machine-outil impose:

$$\text{obj}/(\text{obj}/h)=h$$

$$\frac{x_1}{35} + \frac{x_2}{45} + \frac{x_3}{20} \leq 200$$

- Celle de la main d'oeuvre:

$$\min \rightarrow 4x_1 + 3x_2 + 2x_3 \leq h \rightarrow (170 * 3) * 60$$

De plus:

$$x_1, x_2, x_3 \geq 0$$

$$\max\{60x_1 + 40x_2 + 80x_3\}$$

$$\text{sc} \begin{cases} x_1 & \leq 4900 \\ x_2 & \leq 5400 \\ x_3 & \leq 2000 \\ \frac{x_1}{35} + \frac{x_2}{45} + \frac{x_3}{20} & \leq 200 \Leftrightarrow 36x_1 + 28x_2 + 63x_3 \leq 252000 \\ x_1, x_2, x_3 & \geq 0 \end{cases}$$

Comme $x_1, x_2, x_3 \geq 0$,

$$36x_1 + 27x_2 + 18x_3 \leq 36x_1 + 28x_2 + 63x_3$$

Donc si la 2e est $\leq 252\ 000$, la 1ere est $\leq 252000 \leq 275\ 400$.

Algo du simplexe

III - Séparation et Evaluation

