MOD 4.4: Recherche opérationnelle

Nicolas Bousquet

Ecole Centrale de Lyon

Recherche opérationnelle

Définition (wikipedia)

La Recherche Opérationnelle peut être définie comme l'ensemble des méthodes et techniques rationnelles orientées vers la recherche du meilleur choix dans la façon d'opérer en vue d'aboutir au résultat visé ou au meilleur résultat possible.

Dans la vraie vie:

- Modélisation d'un problème sous forme mathématique.
- Développer des algorithmes de résolution efficace pour ce type de problèmes (ou utiliser des méthodes existantes).
- Evaluer (empiriquement ou théoriquement) la qualité de la solution comparée à la solution optimale.
- Evaluer la solution dans l'environnement réel.
  Toutes les contraintes ont-elles été prises en compte?

Recherche Opérationnelle (cont.)

Le but d'une solution d'un problème en Recher Opérationnelle est que la solution trouvée soit à la fois:

- Efficace. Solution rapide au problème.
- Robuste. Si les données du problème change légèrement (ou que les données obtenus sont bruitées), on doit pouvoir garantir que la solution reste proche d'une solution optimale.
- Garantie théorique ou expérimentale. La solution obtenue doit être la meilleure possible. Si possible optimale, sinon proche de l'être.

Exemples:

- Ordonnancement.
- Routage.
- Optimisation industrielle.
- Aide à la décision.
- Logistique

Exemples:

- Ordonnancement.
- Routage.
- Optimisation industrielle.
- Aide à la décision.
- Logistique

Différentes approches:

- Solutions exactes. (Algo. polynomiaux...)
- Solutions approchées.
- Heuristiques (local search, deep learning...).

Exemples:

- Ordonnancement.
- Routage.
- Optimisation industrielle.
- Aide à la décision.
- Logistique

Différents types de contraintes:

- Discrètes.
- Continues.
- Mixtes.

Différentes approches:

- Solutions exactes. (Algo. polynomiaux...)
- Solutions approchées.
- Heuristiques (local search, deep learning...).

Organisation des deux prochains cours

- Partie 1 - Modélisation.
- Partie 2 - Programmation Linéaire.
- Partie 3 - Dualité et toutes ses conséquences.
- Partie 4 - Programmation Linéaire en nombre entiers.

Modélisation

Définition

La modélisation consiste à représenter mathématiquement un problème réel via des objectifs et des contraintes définis comme des problèmes mathématiques.

Modélisation

Définition

La modélisation consiste à représenter mathématiquement un problème réel via des objectifs et des contraintes définis comme des problèmes mathématiques.

Un procédé qui nécessite souvent plusieurs allers-retours:

- Problèmes mal posés.
- Quelles sont les “vraies” contraintes.
- Contraintes inconnues.

Modélisation 1 - Ordonnancement

- Un ensemble de tâches $X = \{x_1, \ldots, x_n\}$ qui peuvent être effectués par une machine.
- Pour chaque tâche x_i, il existe un intervalle (a_i, b_i) durant lequel la tâche doit être nécessairement être effectuée intégralement.
- Objectif: Maximiser le nombre de tâches effectuées complètement?

Modélisation 1 - Ordonnancement

- Un ensemble de tâches $X = \{x_1, \ldots, x_n\}$ qui peuvent être effectués par une machine.
- Pour chaque tâche x_i, il existe un intervalle (a_i, b_i) durant lequel la tâche doit être nécessairement être effectuée intégralement.
- Objectif: Maximiser le nombre de tâches effectuées complètement?

Kit clé en main pour la modélisation:

- Sur quoi on optimise? $\rightarrow$ Variables.

Modélisation 1 - Ordonnancement

- Un ensemble de tâches $X = \{x_1, \ldots, x_n\}$ qui peuvent être effectués par une machine.
- Pour chaque tâche x_i, il existe un intervalle (a_i, b_i) durant lequel la tâche doit être nécessairement être effectuée intégralement.
- Objectif: Maximiser le nombre de tâches effectuées complètement?

Kit clé en main pour la modélisation:

- Sur quoi on optimise? $\rightarrow$ Variables.

- Exprimer l'objectif en fonction des variables.

Modélisation 1 - Ordonnancement

- Un ensemble de tâches $X = \{x_1, \ldots, x_n\}$ qui peuvent être effectués par une machine.
- Pour chaque tâche x_i, il existe un intervalle (a_i, b_i) durant lequel la tâche doit être nécessairement être effectuée intégralement.
- Objectif: Maximiser le nombre de tâches effectuées complètement?

Kit clé en main pour la modélisation:

- Sur quoi on optimise? $\rightarrow$ Variables.
- Exprimer l'objectif en fonction des variables.
- Exprimer les contraintes en fonction des variables.

Modélisation 1 - Ordonnancement

- Un ensemble de tâches $X = \{x_1, \ldots, x_n\}$ qui peuvent être effectués par une machine.
- Pour chaque tâche x_i, il existe un intervalle (a_i, b_i) durant lequel la tâche doit être nécessairement être effectuée intégralement.
- Objectif: Maximiser le nombre de tâches effectuées complètement?

Kit clé en main pour la modélisation:

- Sur quoi on optimise? $\rightarrow$ Variables.
  $\rightarrow$ Ici on crée, une variable par tâche $x_i \in \{0, 1\}$.
- Exprimer l'objectif en fonction des variables.
- Exprimer les contraintes en fonction des variables.

Modélisation 1 - Ordonnancement

- Un ensemble de tâches $X = \{x_1, \ldots, x_n\}$ qui peuvent être effectués par une machine.
- Pour chaque tâche x_i, il existe un intervalle (a_i, b_i) durant lequel la tâche doit être nécessairement être effectuée intégralement.
- Objectif: Maximiser le nombre de tâches effectuées complètement?

Kit clé en main pour la modélisation:

- Sur quoi on optimise? $\rightarrow$ Variables.
  $\rightarrow$ Ici on crée, une variable par tâche $x_i \in \{0, 1\}$.
- Exprimer l'objectif en fonction des variables.
  $\rightarrow$ Maximize $\sum_{i \leq n} x_i$.
- Exprimer les contraintes en fonction des variables.

Modélisation 1 - Ordonnancement

- Un ensemble de tâches $X = \{x_1, \dots, x_n\}$ qui peuvent être effectués par une machine.
- Pour chaque tâche x_i, il existe un intervalle (a_i, b_i) durant lequel la tâche doit être nécessairement être effectuée intégralement.
- Objectif: Maximiser le nombre de tâches effectuées complètement?

Kit clé en main pour la modélisation:

- Sur quoi on optimise? $\rightarrow$ Variables.
  $\rightarrow$ Ici on crée, une variable par tâche $x_i \in \{0, 1\}$.
- Exprimer l'objectif en fonction des variables.
  $\rightarrow$ Maximize $\sum_{i \leq n} x_i$.
- Exprimer les contraintes en fonction des variables.
  $\rightarrow$ Quelque soit l'instant t, $\sum_{[a_i, b_i] \text{ contiennent } t} x_i \leq 1$.

Ordonnancement (cont.)

Bilan:

$$\max \sum_{i \leq n} x_i$$

soumis à:

$$\sum_{[a_i, b_i] \text{ contiennent } t} x_i \leq 1 \qquad \forall t$$

$$x_i \in \{0, 1\} \qquad \forall i \in X$$

Modélisation 2 - Problème de transport

- n entrepôts $X = \{x_1, \ldots, x_n\}$ t.q. chaque entrepôt x_i a une quantité c_i de ressource.
- m clients $Y = \{y_1, \ldots, y_m\}$ t.q. chaque y_j a un besoin d_j en la ressource.
- Pour chaque paire i, j, il y a un coût de transport $c_{ij} \in \mathbb{R}$ de x_i à y_j par unité.
- Objectif: Satisfaire chaque client à un coût minimum.

Modélisation 2 - Problème de transport

- n entrepôts $X = \{x_1, \ldots, x_n\}$ t.q. chaque entrepôt x_i a une quantité c_i de ressource.
- m clients $Y = \{y_1, \ldots, y_m\}$ t.q. chaque y_j a un besoin d_j en la ressource.
- Pour chaque paire i, j, il y a un coût de transport $c_{ij} \in \mathbb{R}$ de x_i à y_j par unité.
- Objectif: Satisfaire chaque client à un coût minimum.

erdf
L'ÉLECTRICITÉ EN RÉSEAU

Variables: Quantité z_{ij} de ressources qui vont de l'entrepôt x_i au client y_i.

Contraintes:

$$\sum_{j \leq m} z_{ij} \leq c_i \quad \forall i$$

$$\sum_{i \leq n} z_{ij} = d_j \quad \forall j$$

Objectif: $\min \sum_{i,j} c_{ij} z_{ij}$.

Bilan

Bilan:

$$\min \sum_{i,j} c_{ij} z_{ij}$$

soumis à:

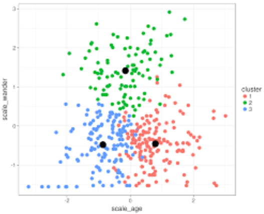
$$\sum_{j \leq m} z_{ij} \leq c_i \qquad \forall i$$

$$\sum_{i \leq n} z_{ij} = d_j \qquad \forall j$$

$$z_{ij} \in \mathbb{N} \qquad \forall i, j$$

Modélisation 3 - Clustering

Un entreprise souhaite ouvrir k entrepôts Y pour servir ses n clients $X = \{ x_1, \dots, x_n \}$ situés sur les points $y_i = (a_i, b_i)$ du plan.
Le but est de minimiser la distance $\mathcal{L}_2$ entre chaque client et son entrepôt le plus proche.



Modélisation 3 - Clustering

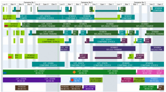
Un entreprise souhaite ouvrir k entrepôts Y pour servir ses n clients $X = \{ x_1, \ldots, x_n \}$ situés sur les points $y_i = (a_i, b_i)$ du plan.
Le but est de minimiser la distance $\mathcal{L}_2$ entre chaque client et son entrepôt le plus proche.

Variables: x_j, y_j pour $j \leq k$.

Objectif: $\min \sum_{i=1}^{n} \min_{j \leq k} d_2(x_i, y_j)$

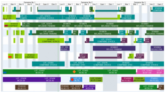
Contraintes: Aucune

Modélisation 4 - Emploi du temps



- Un ensemble de cours $\mathcal{C}$ et une fonction de contrainte f qui dit si deux cours peuvent être données simultanement.
- Un ensemble de k créneaux de cours.
- Trouver une assignation des cours aux différents créneaux qui satisfait toutes les contraintes.

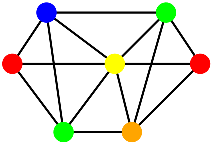
Modélisation 4 - Emploi du temps



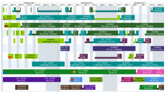
- Un ensemble de cours $\mathcal{C}$ et une fonction de contrainte f qui dit si deux cours peuvent être données simultanement.
- Un ensemble de k créneaux de cours.
- Trouver une assignation des cours aux différents créneaux qui satisfait toutes les contraintes.

Modélisation via un graphe de conflit:

- Sommets: Cours.
- Arêtes entre deux cours si les deux cours sont incompatibles.
- Objectif: Colorer le graphe avec k couleurs en évitant les arêtes monochromatiques.



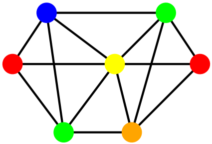
Modélisation 4 - Emploi du temps



- Un ensemble de cours $\mathcal{C}$ et une fonction de contrainte f qui dit si deux cours peuvent être données simultanement.
- Un ensemble de k créneaux de cours.
- Trouver une assignation des cours aux différents créneaux qui satisfait toutes les contraintes.

Modélisation via un graphe de conflit:

- Sommets: Cours.
- Arêtes entre deux cours si les deux cours sont incompatibles.
- Objectif: Colorer le graphe avec k couleurs en évitant les arêtes monochromatiques.



Exercice (difficile): Modéliser ce problème avec variables et contraintes

Programmation Linéaire (PL)

Définition (programme linéaire)

Minimisation/ maximisation d'une fonction linéaire sous des contraintes elles-même linéaires.

Programmation Linéaire (PL)

Définition (programme linéaire)

Minimisation/ maximisation d'une fonction linéaire sous des contraintes elles-même linéaires.

Classiquement:

n = nombre de variables.

$x_1, \ldots, x_n$ = ensemble des variables.

m = nombre de contraintes.

Programmation Linéaire (PL)

Définition (programme linéaire)

Minimisation/ maximisation d'une fonction linéaire sous des contraintes elles-même linéaires.

Classiquement:
n = nombre de variables.
$x_1, \ldots, x_n$ = ensemble des variables.
m = nombre de contraintes.

Exemple:

$$\sum_{i=1}^{n} a_i x_i \leq b$$

Programmation Linéaire (PL)

Définition (programme linéaire)

Minimisation/ maximisation d'une fonction linéaire sous des contraintes elles-même linéaires.

Classiquement:

n = nombre de variables.

$x_1, \ldots, x_n$ = ensemble des variables.

m = nombre de contraintes.

Exemple:

$$\sum_{i=1}^{n} a_i x_i \leq b$$

$$\sum_{i=1}^{n} a_i x_i \geq b \Leftrightarrow \sum_{i=1}^{n} -a_i x_i \leq -b$$

Programmation Linéaire (PL)

Définition (programme linéaire)

Minimisation/ maximisation d'une fonction linéaire sous des contraintes elles-même linéaires.

Classiquement:

n = nombre de variables.

$x_1, \ldots, x_n$ = ensemble des variables.

m = nombre de contraintes.

Exemple:

$$\sum_{i=1}^{n} a_i x_i \leq b$$

$$\sum_{i=1}^{n} a_i x_i \geq b \Leftrightarrow \sum_{i=1}^{n} -a_i x_i \leq -b$$

$$\sum_{i=1}^{n} a_i x_i = b_i \Leftrightarrow \Big((\sum_{i=1}^{n} a_i x_i \leq b) \;\; AND \;\; \sum_{i=1}^{n} a_i x_i \geq b \Big)$$

Programmation Linéaire (PL)

Définition (programme linéaire)

Minimisation/ maximisation d'une fonction linéaire sous des contraintes elles-même linéaires.

Classiquement:

n = nombre de variables.

$x_1, \ldots, x_n$ = ensemble des variables.

m = nombre de contraintes.

Exemple:

$$\sum_{i=1}^{n} a_i x_i \leq b$$

$$\sum_{i=1}^{n} a_i x_i \geq b \Leftrightarrow \sum_{i=1}^{n} -a_i x_i \leq -b$$

$$\sum_{i=1}^{n} a_i x_i = b_i \Leftrightarrow \Big((\sum_{i=1}^{n} a_i x_i \leq b) \;\; AND \;\; \sum_{i=1}^{n} a_i x_i \geq b \Big)$$

$$x_i \geq 0$$

Programmation Linéaire (PL)

Définition (programme linéaire)

Minimisation/ maximisation d'une fonction linéaire sous des contraintes elles-même linéaires.

Classiquement:
n = nombre de variables.
$x_1, \ldots, x_n$ = ensemble des variables.
m = nombre de contraintes.

Exemple:

$$\sum_{i=1}^{n} a_i x_i \leq b$$

$$\sum_{i=1}^{n} a_i x_i \geq b \Leftrightarrow \sum_{i=1}^{n} -a_i x_i \leq -b$$

$$\sum_{i=1}^{n} a_i x_i = b_i \Leftrightarrow \Big((\sum_{i=1}^{n} a_i x_i \leq b) \;\; AND \;\; \sum_{i=1}^{n} a_i x_i \geq b \Big)$$

$$x_i \geq 0$$

Questions: Quels exemples sont exprimés comme des PL?

Forme normale d'un PL

$$\begin{aligned} & \max \sum c_i x_i \\ \text{soumis à} \qquad & \forall j, \sum_{i=1}^{n} a_i x_i \leq b_j \\ & \forall i, x_i \geq 0 \end{aligned}$$

Forme normale d'un PL

$$\begin{aligned} & \max \sum c_i x_i \\ \text{soumis à} \qquad & \forall j, \sum_{i=1}^{n} a_i x_i \leq b_j \\ & \forall i, x_i \geq 0 \end{aligned}$$

A = Matrice des $a_{i,j}$.
c = vecteur des c_i.
b = vecteur des b_i.

Un programme linéaire est sous forme normale s'il s'écrit:

$$\begin{aligned} & \max c^t x \\ \text{soumis à} \qquad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

La contrainte $x \geq 0$ s'appelle contrainte de positivité.

Exemple

Une usine fabrique des chaises et des tables. Chaque semaine l'usine commande la même quantité de bois, connaît le temps de travail de ses employés et la durée de fonctionnement de ses machines.
Sachant qu'une table rapporte 6 euros et une chaise 4, quel est le mix chaises-tables qui maximise le revenu de l'usine?

Exemple

Une usine fabrique des chaises et des tables. Chaque semaine l'usine commande la même quantité de bois, connaît le temps de travail de ses employés et la durée de fonctionnement de ses machines.
Sachant qu'une table rapporte 6 euros et une chaise 4, quel est le mix chaises-tables qui maximise le revenu de l'usine?

	Table	**Chaise**	**Quantité disponible**
Equipement	3	9	81
Main d'oeuvre	4	5	55
Bois	2	1	20

Mise en équation

	Table	**Chaise**	**Quantité disponible**
Equipement	3	9	81
Main d'oeuvre	4	5	55
Bois	2	1	20

Mise en équation

	Table	**Chaise**	**Quantité disponible**
Equipement	3	9	81
Main d'oeuvre	4	5	55
Bois	2	1	20

- Création de deux variables: x_t et x_c.

Mise en équation

	Table	Chaise	Quantité disponible
Equipement	3	9	81
Main d'oeuvre	4	5	55
Bois	2	1	20

- Création de deux variables: x_t et x_c.
- Création de trois contraintes: équipement, main d'oeuvre, bois:

$$\begin{aligned} 3x_t + 9x_c &\leq 81 \\ 4x_t + 5x_c &\leq 55 \\ 2x_t + x_c &\leq 20 \\ x_t, x_c &\geq 0 \end{aligned}$$

Mise en équation

	Table	Chaise	Quantité disponible
Equipement	3	9	81
Main d'oeuvre	4	5	55
Bois	2	1	20

- Création de deux variables: x_t et x_c.
- Création de trois contraintes: équipement, main d'oeuvre, bois:

$$\begin{aligned} 3x_t + 9x_c &\leq 81 \\ 4x_t + 5x_c &\leq 55 \\ 2x_t + x_c &\leq 20 \\ x_t, x_c &\geq 0 \end{aligned}$$

- Création de la fonction objectif.

$$z = \max(6x_t + 4x_c)$$

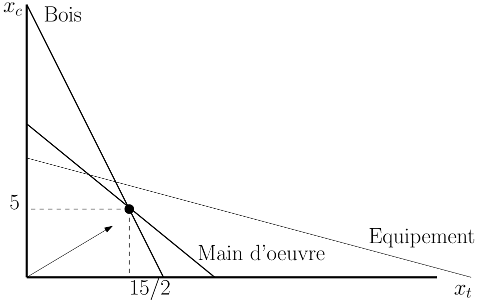
Résolution du système

Quand on résout le système, on obtient la solution optimale suivante:

$$(x_t, x_c) = (15/2, 5).$$

Cette solution peut être obtenue de plusieurs façons:

- En dimension 2: graphiquement.



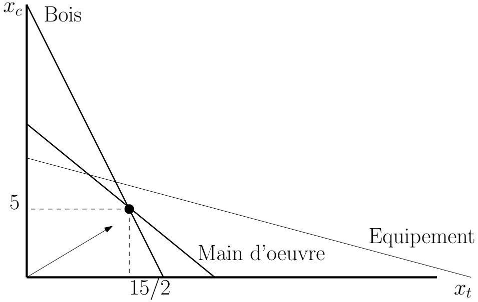
Résolution du système

Quand on résout le système, on obtient la solution optimale suivante:

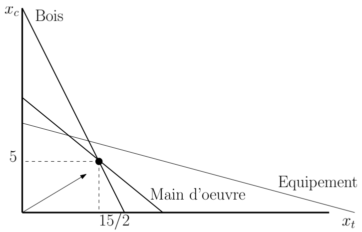
$$(x_t, x_c) = (15/2, 5).$$

Cette solution peut être obtenue de plusieurs façons:

- En dimension 2: graphiquement.
- En toute dimension: comment faire?

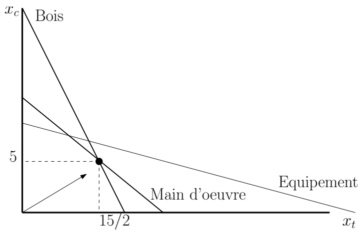


Points extrêmes et recherche de solution optimale



Une contrainte $\sum a_i x_i \leq b$ définit un demi-espace.
La contrainte $\sum a_i x_i \leq b$ est serrée pour un point z si $\sum a_i z_i = b$.

Points extrêmes et recherche de solution optimale



Une contrainte $\sum a_i x_i \leq b$ définit un demi-espace.
La contrainte $\sum a_i x_i \leq b$ est serrée pour un point z si $\sum a_i z_i = b$.

Un point extrême z est un point tel que:

- z soit une solution,
- z est l'unique point à l'intersection d'un ensemble $\mathcal{C}$ de contraintes serrées.

A quoi ressemble une solution optimale?

Remarque:
L'ensemble des solutions est une intersection de demi-espaces fermés.
En particulier c'est un espace convexe et fermé.

A quoi ressemble une solution optimale?

Remarque:
L'ensemble des solutions est une intersection de demi-espaces fermés. En particulier c'est un espace convexe et fermé.

Théorème 1:
On optimise une fonction linéaire (donc convexe et concave) dans un convexe
$\Rightarrow$ Un maximum (resp. minimum) local est un maximum (resp. minimum) global.

A quoi ressemble une solution optimale?

Remarque:
L'ensemble des solutions est une intersection de demi-espaces fermés. En particulier c'est un espace convexe et fermé.

Théorème 1:
On optimise une fonction linéaire (donc convexe et concave) dans un convexe
$\Rightarrow$ Un maximum (resp. minimum) local est un maximum (resp. minimum) global.

Théorème 2:
Si la valeur optimale est finie alors il existe un point extrême qui atteint le maximum.

A quoi ressemble une solution optimale?

Remarque:
L'ensemble des solutions est une intersection de demi-espaces fermés.
En particulier c'est un espace convexe et fermé.

Théorème 1:
On optimise une fonction linéaire (donc convexe et concave) dans un convexe
$\Rightarrow$ Un maximum (resp. minimum) local est un maximum (resp. minimum) global.

Théorème 2:
Si la valeur optimale est finie alors il existe un point extrême qui atteint le maximum.

Principe de l'algorithme du simplexe: Se promener de points extrêmes en points extrêmes.

Etape 1: Mettre le PL sous forme standard

Forme standard:

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

Etape 1: Mettre le PL sous forme standard

Forme standard:

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

Idée: rajouter des variables d'écart.

$$\sum_{i=1}^{n} a_i x_i \leq b$$

Créer une variable y et poser:

$$y = b - \sum_{i=1}^{n} a_i x_i$$

Notez que $y \geq 0$ et $\sum_{i=1}^{n} a_i x_i + y = b$.

Remarque: La forme générale des solutions ne change pas. La projection d'une solution est toujours une solution.

Sur notre exemple

- On a les inégalités.

$$\begin{aligned} 3x_t + 9x_c &\leq 81 \\ 4x_t + 5x_c &\leq 55 \\ 2x_t + x_c &\leq 20 \\ x_t, x_c &\geq 0 \end{aligned}$$

Sur notre exemple

- On a les inégalités.

$$\begin{aligned} 3x_t + 9x_c &\le 81 \\ 4x_t + 5x_c &\le 55 \\ 2x_t + x_c &\le 20 \\ x_t, x_c &\ge 0 \end{aligned}$$

- On rajoute les variables d'écart pour créer des égalités.

$$z = \max 6x_t + 4x_c$$

sous les contraintes

$$\begin{aligned} 3x_t + 9x_c + y_1 &= 81 \\ 4x_t + 5x_c + y_2 &= 55 \\ 2x_t + x_c + y_3 &= 20 \\ x_t, x_c, y_1, y_2, y_3 &\ge 0 \end{aligned} \quad \Leftrightarrow \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix} = \begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Etape 2: Transformer la solution courante

- On cherche une solution réalisable pour commencer:

$$\Leftrightarrow \begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

Autrement dit *"Quand x_t et x_c valent* 0 *alors* $(y_1, y_2, y_3) = (81, 55, 20)$ *est une solution"*.

Etape 2: Transformer la solution courante

- On cherche une solution réalisable pour commencer:

$$\Leftrightarrow \begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

  Autrement dit *"Quand x_t et x_c valent 0 alors $(y_1, y_2, y_3) = (81, 55, 20)$ est une solution"*.

- Notre solution est-elle améliorable?

$$\max 6x_t + 4x_c$$

  Pour l'instant x_t et x_c valent 0... Donc si on les rend positif, ça améliorera la valeur de la solution !

Etape 2: Transformer la solution courante

- On cherche une solution réalisable pour commencer:

$$\Leftrightarrow \begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

  Autrement dit *"Quand x_t et x_c valent* 0 *alors* $(y_1, y_2, y_3) = (81, 55, 20)$ *est une solution"*.

- Notre solution est-elle améliorable?

$$\max 6x_t + 4x_c$$

  Pour l'instant x_t et x_c valent 0... Donc si on les rend positif, ça améliorera la valeur de la solution !

- On choisit une variable avec un coefficient positif dans l'objectif et on l'augmente "autant qu'on peut" (en maintenant les contraintes). $\rightarrow$ Choisissons arbitrairement x_t.

Test d'entrée

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

- Si x_t est modifié (et x_c reste égal à 0) alors on doit toujours avoir:

$$3x_t + y_1 = 81$$
$$4x_t + y_2 = 55$$
$$2x_t + y_3 = 20$$

Test d'entrée

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

- Si x_t est modifié (et x_c reste égal à 0) alors on doit toujours avoir:

$$3x_t + y_1 = 81$$
$$4x_t + y_2 = 55$$
$$2x_t + y_3 = 20$$

- Comme on doit toujours avoir y_1, y_2, y_3 positifs ou nuls on a:

$$x_t \leq 17 \qquad \text{contrainte 1}$$
$$x_t \leq \frac{55}{4} \qquad \text{contrainte 2}$$
$$x_t \leq 10 \qquad \text{contrainte 3}$$

Test d'entrée

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

- Si x_t est modifié (et x_c reste égal à 0) alors on doit toujours avoir:

$$3x_t + y_1 = 81$$
$$4x_t + y_2 = 55$$
$$2x_t + y_3 = 20$$

- Comme on doit toujours avoir y_1, y_2, y_3 positifs ou nuls on a:

$$x_t \leq 17 \quad \text{contrainte 1}$$
$$x_t \leq \frac{55}{4} \quad \text{contrainte 2}$$
$$x_t \leq 10 \quad \text{contrainte 3}$$

- Plus on augmente x_t mieux c'est (pour l'objectif) $\rightarrow$, on augmente x_t jusqu'à 10. A ce moment là, y_3 devient égal à 0.
  On dit alors que y_3 sort de la base et que x_t y rentre.

Variables basiques et hors base

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

- On remarque la matrice restreinte aux y_i était l'identité.
  On supposera dorénavant que la matrice restreinte aux variables dans la base (ici les y_i) est l'identité.
  Les variables dans la base sont les seules qui **pourront être non nulles** dans la solution courante (ici $(0, 0, 81, 55, 20)$).

Variables basiques et hors base

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

- On remarque la matrice restreinte aux y_i était l'identité.
  On supposera dorénavant que la matrice restreinte aux variables dans la base (ici les y_i) est l'identité.
  Les variables dans la base sont les seules qui **pourront être non nulles** dans la solution courante (ici $(0, 0, 81, 55, 20)$).
- Maintenant on ajouter x_t dans la base et en supprimer y_3.
  $\rightarrow$ Elimination de Jordan Gauss pour que la matrice restreinte aux colonnes de x_t, y_1, y_2 soit l'identité.

Variables basiques et hors base

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

- On remarque la matrice restreinte aux y_i était l'identité.
  On supposera dorénavant que la matrice restreinte aux variables dans la base (ici les y_i) est l'identité.
  Les variables dans la base sont les seules qui **pourront être non nulles** dans la solution courante (ici $(0, 0, 81, 55, 20)$).
- Maintenant on ajouter x_t dans la base et en supprimer y_3.
  $\rightarrow$ Elimination de Jordan Gauss pour que la matrice restreinte aux colonnes de x_t, y_1, y_2 soit l'identité.

$$\begin{pmatrix} 0 & \frac{15}{2} & 1 & 0 & -\frac{3}{2} \\ 0 & 3 & 0 & 1 & -2 \\ 1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 51 \\ 15 \\ 10 \end{pmatrix}$$

Variables basiques et hors base

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

- On remarque la matrice restreinte aux y_i était l'identité.
  On supposera dorénavant que la matrice restreinte aux variables dans la base (ici les y_i) est l'identité.
  Les variables dans la base sont les seules qui **pourront être non nulles** dans la solution courante (ici $(0, 0, 81, 55, 20)$).
- Maintenant on ajouter x_t dans la base et en supprimer y_3.
  $\rightarrow$ Elimination de Jordan Gauss pour que la matrice restreinte aux colonnes de x_t, y_1, y_2 soit l'identité.

$$\begin{pmatrix} 0 & \frac{15}{2} & 1 & 0 & -\frac{3}{2} \\ 0 & 3 & 0 & 1 & -2 \\ 1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 51 \\ 15 \\ 10 \end{pmatrix}$$

Nouvelle solution et objectifs

$$\begin{pmatrix} 0 & \frac{15}{2} & 1 & 0 & -\frac{3}{2} \\ 0 & 3 & 0 & 1 & -2 \\ 1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 51 \\ 15 \\ 10 \end{pmatrix}$$

- Nouvelle solution courante: $(10, 0, 51, 15, 0)$.
- On re-exprime la fonction objectif en fonction des variables hors base: $6x_t + 4x_c = (60 - 3x_c - 3y_3) + 4x_c = 60 + x_c - 3y_3$.

Nouvelle solution et objectifs

$$\begin{pmatrix} 0 & \frac{15}{2} & 1 & 0 & -\frac{3}{2} \\ 0 & 3 & 0 & 1 & -2 \\ 1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 51 \\ 15 \\ 10 \end{pmatrix}$$

- Nouvelle solution courante: $(10, 0, 51, 15, 0)$.
- On re-exprime la fonction objectif en fonction des variables hors base: $6x_t + 4x_c = (60 - 3x_c - 3y_3) + 4x_c = 60 + x_c - 3y_3$.

Remarques:

- On a simplement **ré-écrit** la matrice de contraintes !
  $\Rightarrow$ Les solutions du système sont toujours les mêmes. (Autrement dit, si vous pluggez $(0, 0, 81, 55, 20)$ dans ce système, ça vous donnera **toujours** une solution et sa valeur sera toujours 0).

- Similairement on a simplement **ré-écrit** la fonction objectif !

Nouvelle solution et objectifs

$$\begin{pmatrix} 0 & \frac{15}{2} & 1 & 0 & -\frac{3}{2} \\ 0 & 3 & 0 & 1 & -2 \\ 1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 51 \\ 15 \\ 10 \end{pmatrix}$$

- Nouvelle solution courante: $(10, 0, 51, 15, 0)$.
- On re-exprime la fonction objectif en fonction des variables hors base: $6x_t + 4x_c = (60 - 3x_c - 3y_3) + 4x_c = 60 + x_c - 3y_3$.

Remarques:

- On a simplement **ré-écrit** la matrice de contraintes !
  $\Rightarrow$ Les solutions du système sont toujours les mêmes. (Autrement dit, si vous pluggez $(0, 0, 81, 55, 20)$ dans ce système, ça vous donnera **toujours** une solution et sa valeur sera toujours 0).
- Mais à chaque matrice on associe une solution appelée *basic feasible solution* qui correspond à "variables de la base" potentiellement non nulle et les autres sont nulles.
- Similairement on a simplement **ré-écrit** la fonction objectif !

Nouvelle solution et objectifs

$$\max 60 + x_c - 3y_3$$

soumis à

$$\begin{pmatrix} 0 & \frac{15}{2} & 1 & 0 & -\frac{3}{2} \\ 0 & 3 & 0 & 1 & -2 \\ 1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 51 \\ 15 \\ 10 \end{pmatrix}$$

Exercice:

1. Qui peut rentrer dans la base?
2. Qui doit en sortir?
3. Calculer la nouvelle matrice de contrainte et la nouvelle fonction objectif.

Solution optimale !

$$z = 65 - \frac{1}{3}y_2 - \frac{7}{3}y_3$$

sous les contraintes:

$$\begin{pmatrix} 0 & 0 & 1 & -5/2 & 7/2 \\ 0 & 1 & 0 & 1/3 & -2/3 \\ 1 & 0 & 0 & -1/6 & 5/6 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 27/2 \\ 5 \\ 15/2 \end{pmatrix}$$

Pour répondre à vos questions:

Solution optimale !

$$z = 65 - \frac{1}{3}y_2 - \frac{7}{3}y_3$$

sous les contraintes:

$$\begin{pmatrix} 0 & 0 & 1 & -5/2 & 7/2 \\ 0 & 1 & 0 & 1/3 & -2/3 \\ 1 & 0 & 0 & -1/6 & 5/6 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 27/2 \\ 5 \\ 15/2 \end{pmatrix}$$

Pour répondre à vos questions:

- Quelle est la solution? Il s'agit du point $(15/2, 5, 25/2, 0, 0)$.

Solution optimale !

$$z = 65 - \frac{1}{3}y_2 - \frac{7}{3}y_3$$

sous les contraintes:

$$\begin{pmatrix} 0 & 0 & 1 & -5/2 & 7/2 \\ 0 & 1 & 0 & 1/3 & -2/3 \\ 1 & 0 & 0 & -1/6 & 5/6 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 27/2 \\ 5 \\ 15/2 \end{pmatrix}$$

Pour répondre à vos questions:

- Quelle est la solution? Il s'agit du point $(15/2, 5, 25/2, 0, 0)$.
- Quelle est sa valeur? 65.

Solution optimale !

$$z = 65 - \frac{1}{3}y_2 - \frac{7}{3}y_3$$

sous les contraintes:

$$\begin{pmatrix} 0 & 0 & 1 & -5/2 & 7/2 \\ 0 & 1 & 0 & 1/3 & -2/3 \\ 1 & 0 & 0 & -1/6 & 5/6 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 27/2 \\ 5 \\ 15/2 \end{pmatrix}$$

Pour répondre à vos questions:

- Quelle est la solution? Il s'agit du point $(15/2, 5, 25/2, 0, 0)$.
- Quelle est sa valeur? 65.
- Pourquoi la solution est optimale? Le mieux que l'on puisse faire c'est d'avoir y_2 et y_3 équaux à 0... Et c'est justement le cas dans la solution courante !

Solution optimale !

$$z = 65 - \frac{1}{3}y_2 - \frac{7}{3}y_3$$

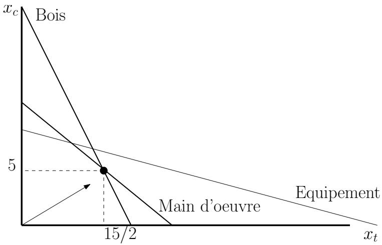
sous les contraintes:

$$\begin{pmatrix} 0 & 0 & 1 & -5/2 & 7/2 \\ 0 & 1 & 0 & 1/3 & -2/3 \\ 1 & 0 & 0 & -1/6 & 5/6 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 27/2 \\ 5 \\ 15/2 \end{pmatrix}$$

Pour répondre à vos questions:

- Quelle est la solution? Il s'agit du point $(15/2, 5, 25/2, 0, 0)$.
- Quelle est sa valeur? 65.
- Pourquoi la solution est optimale? Le mieux que l'on puisse faire c'est d'avoir y_2 et y_3 équaux à 0... Et c'est justement le cas dans la solution courante !
- Mais pourquoi ça termine? Et pourquoi ça marcherait à chaque fois? Et comment interpréter ce qu'on vient de faire? Et comment on fait pour commencer?

Interprétation géométrique



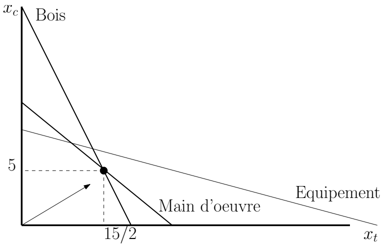
Au départ:

$$\max 6x_t + 4x_c$$

soumis à

$$\begin{pmatrix} 3 & 9 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 81 \\ 55 \\ 20 \end{pmatrix}$$

Interprétation géométrique



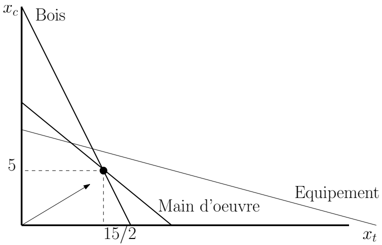
A l'étape intermédiaire:

$$\max 60 + x_c - 3y_3$$

soumis à

$$\begin{pmatrix} 0 & \frac{15}{2} & 1 & 0 & -\frac{3}{2} \\ 0 & 3 & 0 & 1 & -2 \\ 1 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 51 \\ 15 \\ 10 \end{pmatrix}$$

Interprétation géométrique



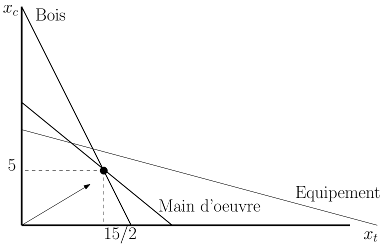
A la fin:

$$z = 65 - \frac{1}{3}y_2 - \frac{7}{3}y_3$$

soumis à:

$$\begin{pmatrix} 0 & 0 & 1 & -5/2 & 7/2 \\ 0 & 1 & 0 & 1/3 & -2/3 \\ 1 & 0 & 0 & -1/6 & 5/6 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 27/2 \\ 5 \\ 15/2 \end{pmatrix}$$

Interprétation géométrique



A la fin:

$$z = 65 - \frac{1}{3}y_2 - \frac{7}{3}y_3$$

soumis à:

$$\begin{pmatrix} 0 & 0 & 1 & -5/2 & 7/2 \\ 0 & 1 & 0 & 1/3 & -2/3 \\ 1 & 0 & 0 & -1/6 & 5/6 \end{pmatrix} \begin{pmatrix} x_t \\ x_c \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 27/2 \\ 5 \\ 15/2 \end{pmatrix}$$

Théorème: Le Simplexe consiste à se "promener" de points extrêmes en points extrêmes en améliorant localement la solution !

Et pourquoi ça termine?

Remarque: L'algorithme du simplexe est en fait une classe d'algorithme qui dépend de la règle de pivot choisie (choix de la variable qui rentre et qui sort).

Et pourquoi ça termine?

Remarque: L'algorithme du simplexe est en fait une classe d'algorithme qui dépend de la règle de pivot choisie (choix de la variable qui rentre et qui sort).

Théorème 1: Si il n'existe pas $(n + 1)$ droites qui s'intersectent en un même point (système non dégénéré), alors on améliore strictement la solution à chaque étape et l'algorithme termine.

Et pourquoi ça termine?

Remarque: L'algorithme du simplexe est en fait une classe d'algorithme qui dépend de la règle de pivot choisie (choix de la variable qui rentre et qui sort).

Théorème 1: Si il n'existe pas $(n+1)$ droites qui s'intersectent en un même point (système non dégénéré), alors on améliore strictement la solution à chaque étape et l'algorithme termine.

Dans les autre cas ce n'est pas clair... Ca dépend de quelle variable on choisit pour rentrer et sortir...

Algorithme du Simplexe proposé en 1947 par Dantzig.
Première règle de terminaison: Bland en 1977 !

Et pourquoi ça termine?

Remarque: L'algorithme du simplexe est en fait une classe d'algorithme qui dépend de la règle de pivot choisie (choix de la variable qui rentre et qui sort).

Théorème 1: Si il n'existe pas $(n+1)$ droites qui s'intersectent en un même point (système non dégénéré), alors on améliore strictement la solution à chaque étape et l'algorithme termine.

Dans les autre cas ce n'est pas clair... Ca dépend de quelle variable on choisit pour rentrer et sortir...

Algorithme du Simplexe proposé en 1947 par Dantzig.
Première règle de terminaison: Bland en 1977 !

Théorème 2: (Bland) Il existe une règle (déterministe) qui garantit que le Simplexe termine en temps fini.

Problème ouvert: Existe-t-il une règle (déterministe) qui garantit que le Simplexe termine en temps polynomial?

Complexité

Problème ouvert: Existe-t-il une règle (déterministe) qui garantit que le Simplexe termine en temps polynomial?

Conjecture (Hirsch): Il existe toujours une suite d'étapes de longueur $n + m$ pour aller de n'importe quel point extrême à la solution optimale. Recement réfutée... Mais toujours ouvert pour $n + m + 1$.

Complexité

Problème ouvert: Existe-t-il une règle (déterministe) qui garantit que le Simplexe termine en temps polynomial?

Conjecture (Hirsch): Il existe toujours une suite d'étapes de longueur $n + m$ pour aller de n'importe quel point extrême à la solution optimale.Recement réfutée... Mais toujours ouvert pour $n + m + 1$.

Mais il y a quand même deux bonnes nouvelles:

- En pratique, l'algorithme du Simplexe fonctionne bien ($3/2(n + m)$ itérations en moyenne).

Complexité

Problème ouvert: Existe-t-il une règle (déterministe) qui garantit que le Simplexe termine en temps polynomial?

Conjecture (Hirsch): Il existe toujours une suite d'étapes de longueur $n + m$ pour aller de n'importe quel point extrême à la solution optimale.Recement réfutée... Mais toujours ouvert pour $n + m + 1$.

Mais il y a quand même deux bonnes nouvelles:

- En pratique, l'algorithme du Simplexe fonctionne bien $(3/2(n + m)$ itérations en moyenne).
- En théorie. L'algorithme dit "de l'ellipsoïde" peut résoudre *tous* les programmes linéaires en temps polynomial dans le pire des cas? Mais il est difficile à comprendre, à implémenter et (beaucoup) moins efficace en pratique.

Petite remarque d'importance

Notons que la solution optimale n'est pas entière. Et fabriquer 7.5 tables n'a pas de sens.
Pour "résoudre" le problème il suffit d'*arrondir la solution optimale à l'entier inférieur*, ce qui ne modifie (en général) pas sensiblement la valeur obtenue (surtout quand elle la production est de plusieurs centaines / milliers).

Petite remarque d'importance

Notons que la solution optimale n'est pas entière. Et fabriquer 7.5 tables n'a pas de sens.
Pour "résoudre" le problème il suffit d'*arrondir la solution optimale à l'entier inférieur*, ce qui ne modifie (en général) pas sensiblement la valeur obtenue (surtout quand elle la production est de plusieurs centaines / milliers).

Faire un tel arrondi peut néanmoins se révéler problématique si:

- L'industriel veut réellement une solution optimale (c'est rare).
- Les entiers sont "petits" et arrondir modifie sensiblement la valeur de la solution optimale (ça peut arriver).

Programme Linéaire en nombre entiers (PLNE)

On peut modéliser de nombreux problèmes avec des PL ayant comme variables $\{0, 1\}$. Par exemple:

- Selection de tâches en ordonnancement (Exemple 1).
- Modélisation de formules booléennes.
- Ensemble dans un graphe ayant une "bonne" propriété (voir BE).

Programme Linéaire en nombre entiers (PLNE)

On peut modéliser de nombreux problèmes avec des PL ayant comme variables $\{0, 1\}$. Par exemple:

- Selection de tâches en ordonnancement (Exemple 1).
- Modélisation de formules booléennes.
- Ensemble dans un graphe ayant une “bonne” propriété (voir BE).

$\rightarrow$ De nombreux problèmes sont NP-complets et non approximables.

Programme Linéaire en nombre entiers (PLNE)

On peut modéliser de nombreux problèmes avec des PL ayant comme variables $\{0, 1\}$. Par exemple:

- Selection de tâches en ordonnancement (Exemple 1).
- Modélisation de formules booléennes.
- Ensemble dans un graphe ayant une "bonne" propriété (voir BE).

$\rightarrow$ De nombreux problèmes sont NP-complets et non approximables.

Bilan: Les programmes linéaires en nombre entiers sont *beaucoup* plus compliqués que ceux continus...