Cours 4 :
Le Simplexe,
Les deux phases
et la PLNE

Eric Bourreau

+ Vincent Boudet

L’algorithme du simplexe (vocabulaire)

- Forme canonique
- Forme standard
- Dictionnaire :
  - Il partitionne *m* variables et les exprime en fonction des *n* autres.
  - $D = X_B \cup X_N$ où X_B sont les variables de base, et X_N les variables hors-base.
  - Les variables hors base sont les variables de décision, elles sont libres car on peut leur donner n’importe quelle valeur (dans le domaine défini par les contraintes) alors que les variables de base sont liées aux variables hors-base.
- Une solution consiste à mettre toutes les variables hors bases à 0.
- Une solution du dictionnaire correspond à un sommet du polyèdre.

Algorithme du Simplexe (principe) 1/2

- **Initialisation** : Trouver une solution de base initiale (sommet du polyèdre).
  - Si tous les $b_j \geqslant 0$ alors $\begin{cases} x_i = 0 & i = 1\dots n \\ x_{n+j} = b_j & j = 1\dots m \end{cases}$ est une solution initiale réalisable

    et $(D_0) \begin{cases} z & = & \sum_{i=1}^{n} c_i x_i \\ x_{n+j} & = & b_j - \sum_{i=1}^{n} a_{ij} x_i, \quad j = 1, \dots, m \end{cases}$ est un dictionnaire initial réalisable
  - Sinon utiliser le simplexe à 2 phases ou une autre méthode (cf. plus tard).
  - Si on ne peut pas trouver de dictionnaire réalisable, le problème n'a pas de solution

- **À chaque itération** k : soit $(D_k) \begin{cases} z & = & v + \sum_{x_i \in X_N} c'_i x_i \\ x_j & = & b'_j + \sum_{x_i \in X_N} a'_{ij} x_i, \quad \forall x_j \in X_B \end{cases}$

  le dictionnaire réalisable courant, associé à la **solution de base courante** S_k:
  $(x_i = 0 \; \forall x_i \in X_N, \, x_j = b'_j \; \forall x_j \in X_B, \, z = v)$

  (1) **Déterminer si S_k est optimale** : On cherche parmi les variables hors base de S_k, une variable dont l'augmentation permettrait d'augmenter z

  Variables candidates : $\{x_i \in X_N, \; c'_i \geqslant 0\}$: variables apparaissant dans la définition de z et ayant un **coût réduit** c'_i positif

Algorithme du Simplexe (principe) 2/2

→ **Si** $\{x_i \in X_N,\ c'_i \geqslant 0\} = \varnothing$: c-à-d s'il n'existe plus de variables candidates à l'entrée en base (**tous les coûts réduits sont négatifs**).

$\Longrightarrow$ **STOP : La solution courante S_k est optimale** (on dit aussi que le dictionnaire D_k est optimal)

→ **Sinon**, choisir la variable à augmenter (*i.e.* entrant en base) : x_{i*} tel que

$i^* = \arg\max\{c'_i \geqslant 0, x_i \in X_N\}$

(c-à-d, **la variable hors base dont le coefficient dans z est le plus élevé**)

et aller en (2)

(2) **Définir la variable sortant de base** $x_{j*} \in X_B$: celle qui **s'annule en premier lorsque** x_{i*} **augmente**.

En déduire la **nouvelle solution** S_{k+1}

Si aucune variable ne s'annule alors le problème est non-borné.

(3) **Construire un nouveau dictionnaire** D_{k+1} : x_{i*} entre en base et x_{j*} sort de la base.

Remplacer dans D_k, la variable entrant en base, x_{i*}, par son expression en fonction de x_{j*} et des variables $X_N \backslash x_{i*}$

$$x_{j*} = b'_{j*} + a'_{i*j*} x_{i*} + \sum_{x_i \in X_N \backslash x_{i*}} a'_{ij*} x_i \Rightarrow x_{i*} = -\frac{b'_{j*}}{a'_{i*j*}} + \frac{x_{j*}}{a'_{i*j*}} - \sum_{x_i \in X_N \backslash x_{i*}} \frac{a'_{ij*}}{a'_{i*j*}} x_i$$

(opération de **pivot**)

Vérifier que la solution S_{k+1} se retrouve bien dans le dictionnaire D_{k+1}.

Le retour du kouign amann

- Faire tourner le simplexe à la main pour résoudre

$$\begin{cases} Max \quad 1.25 * 6 * \ x_1 + 2.50 * 8\, x_2 \\ \qquad 500x_1 + 250x_2 \leq 10\,000 \quad (1) \\ \qquad 200x_1 + 200x_2 \leq 10\,000 \quad (2) \end{cases}$$

- $$\begin{array}{c|cc|c} & 1 & 2 & \\ \hline 3 & -2 & -1 & 40 \\ 4 & -1 & -1 & 50 \\ \hline & 3 & 8 & 0 \end{array}$$

  - x_2 entre
  - *argmin*(40,50)
  - $x_3$sort

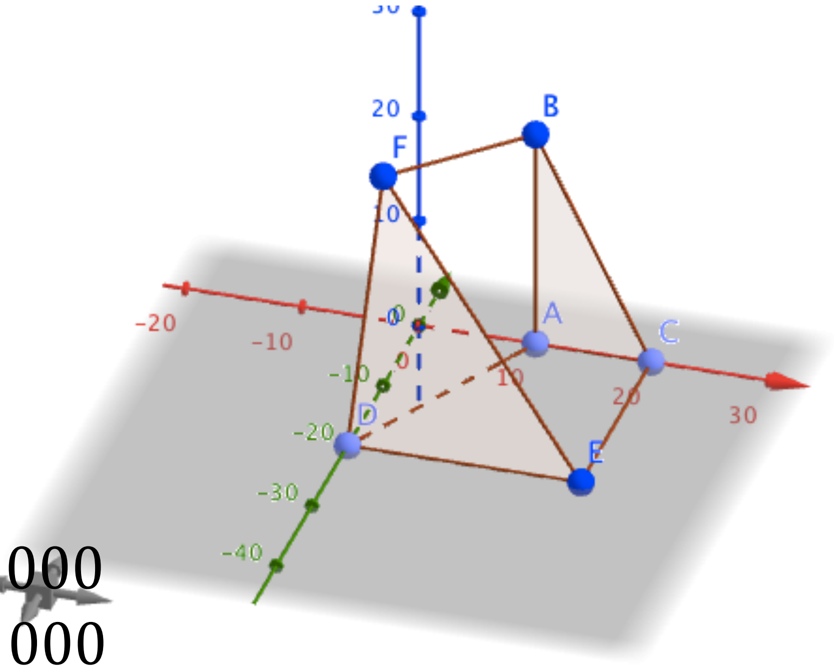
- $$\begin{array}{c|cc|c} & 1 & 3 & \\ \hline 1 & -2 & -1 & 40 \\ 4 & 1 & 1 & 10 \\ \hline & -13 & -1 & 320 \end{array}$$

  optimum

Retour de la rentabilité

- Nouveau PL sans la solution (0,0) admissible pour initialiser le simplexe

$$\begin{cases} Max \quad 7.5\,x_1 + 20\,x_2 \\ 500x_1 + 250x_2 \leq 10\,000 \quad (1) \\ 200x_1 + 200x_2 \leq 10\,000 \quad (2) \\ \qquad avec \;\; x_1 \geq 10 \\ \qquad\qquad\quad x_2 \geq 0 \end{cases}$$



- Programme Linéaire Auxiliaire (PLA)

$$\begin{cases} \quad Max \quad -x_0 \\ 500x_1 + 250x_2 + x_3 \qquad\qquad\qquad\qquad = 10\,000 \\ 200x_1 + 200x_2 + \quad + x_4 \qquad\qquad\quad = 10\,000 \\ \quad -x_1 + \qquad\qquad\qquad\qquad + x_5 - x_0 = \quad -10 \end{cases}$$

Le Simplexe en 2 phases

Phase I : Trouver une solution réalisable

- Rappel variables d'écart : $\forall j = 1, \ldots, m, \sum_{i=1}^{n} a_{ij} x_i \leqslant b_j$ d'où : $\sum_{i=1}^{n} a_{ij} x_i + x_{n+j} = b_j$

  C'est les $b_j < 0$ qui posent problème, car si toutes les variables de décision sont nulles alors les variables d'écart associées $x_{n+j} = b_j < 0$
- Idée : on introduit une nouvelle variable, appelée **variable auxiliaire**, $x_0 \geqslant 0$:

  $\forall j = 1, \ldots, m, \sum_{i=1}^{n} a_{ij} x_i \leqslant b_j + x_0$ de telle sorte que le second terme soit toujours positif

  donc si $x_0 = -\min_j b_j$ alors $\sum_{i=1}^{n} a_{ij} x_i \leqslant b'_j = b_j + x_0$ avec $b'_j \geqslant 0, \ \forall j$ et donc toutes les variables d'écart pourront être positives ou nulles si les variables de décision sont nulles.

Simplexe en 2 phases

Trouver une solution réalisable de $(PL) \iff$ Résoudre un autre problème de type (PL)

$$(PL_{auxilaire}) \begin{cases} \text{Maximiser} & -x_0 & \text{(Minimiser } x_0\text{)} \\ \text{s.c.} & \displaystyle\sum_{i=1}^{n} a_{ij}x_i \leqslant b_j, & \text{pour les } j = 1, \dots, m \text{ tel que } b_j \geqslant 0 \\ \text{et} & \displaystyle\sum_{i=1}^{n} a_{ij}x_i - x_0 \leqslant b_j, & \text{pour les } j = 1, \dots, m \text{ tel que } b_j < 0 \\ \text{et} & x_0 \geqslant 0,\ x_i \geqslant 0, & i = 1, \dots, n \end{cases}$$

Solution réalisable initiale du $(PL_{auxilaire})$

$$\begin{cases} x_i = 0 & i = 1, \dots, n & : \text{variables de décision nulles} \\ x_0 = -\min\limits_{j=1,\dots,m} b_j & & : \text{variable auxiliaire} \\ x_{n+j} = b_j & j = 1, \dots, m \text{ tel que } b_j \geqslant 0 & : \text{variables d'écart} \\ x_{n+j} = b_j + x_0 & j = 1, \dots, m \text{ tel que } b_j < 0 & : \text{variables d'écart} \end{cases}$$

Application

- On considère le programme linéaire suivant :

$$\begin{cases} Max \quad x_1 - 2x_2 + x_3 \\ 2x_1 - x_2 + 2x_3 \leq 4 \\ 2x_1 - 3x_2 + x_3 \leq -5 \\ -x_1 + x_2 - 2x_3 \leq -1 \\ x_1,\ x_2,\ x_3 \geq 0 \end{cases}$$

PLA : (5,0,0,0,4,0,4) -5

x_2<-->x_6 (2,0,1,0,5,0,0) -2

x_3<-->x_0 (0,0,11/5,8/5,3,0,0) 0

PL : (0,11/5,8/5,3,0,0) -14/5

x_6<-->x_4 (0,14/5,17/5,0,0,3) -11/5

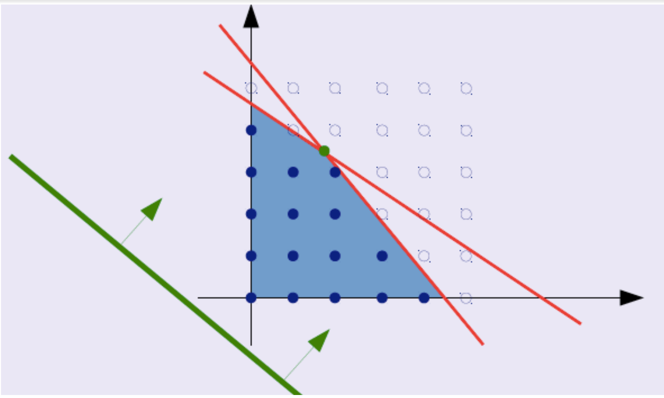
$$\begin{cases} Max \quad z = -x_0 \\ 2x_1 - x_2 + 2x_3 + x_4 = 4 \\ 2x_1 - 3x_2 + x_3 + x_5 - x_0 = -5 \\ -x_1 + x_2 - 2x_3 + x_6 - x_0 = -1 \\ x_0,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6 \geq 0 \end{cases}$$

Programmation Linéaire en Nombres Entiers

Définition : Programmation Linéaire en Nombres Entiers - PLNE

Un problème de PLNE est un problème de PL auquel on a ajouté la contrainte supplémentaire que toutes les variables prennent des valeurs entières (ou booléennes).

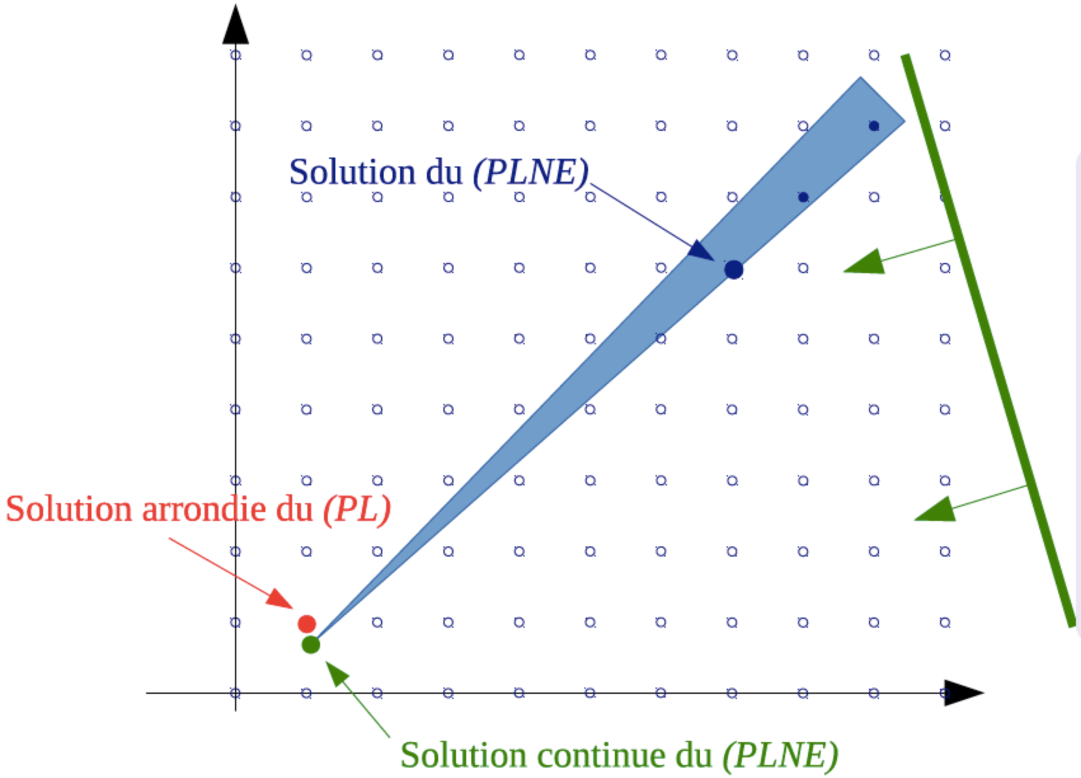
$$(PL)\left\{\begin{array}{ll} \text{Max.} & z(x) = \sum_{i=1}^{n} c_i x_i \\ \text{s.c.} & \sum_{i=1}^{n} a_{ij} x_i \leqslant b_j, \ \forall j = 1, \dots, m \\ \text{et} & x_i \geqslant 0, \qquad \forall i = 1, \dots, n \end{array}\right. \qquad (PLNE)\left\{\begin{array}{ll} \text{Max.} & z(x) = \sum_{i=1}^{n} c_i x_i \\ \text{s.c.} & \sum_{i=1}^{n} a_{ij} x_i \leqslant b_j, \ \forall j = 1, \dots, m \\ \text{et} & x \in \mathbb{N}^n \text{ ou } x \in \{0, 1\}^n \end{array}\right.$$



Relaxation de contraintes

- La relaxation d'une ou plusieurs contraintes consiste à l'allègement voire la suppression de celles ci.
- La relaxation d'un problème d'optimisation *P* est donc un problème d'optimisation *P'* où :
  - La solution x'^* n'est pas nécessairement une solution de P,
  - Dans la cas de la maximisation, $z(x'^*) \geq z(x^*)$, on dit que c'est un majorant,
  - Dans le cas de la minimisation, $z(x'^*) \leq z(x^*)$, on dit que c'est un minorant.
- On appelle saut (ou gap) de relaxation l'écart $|z(x'*) - z(x*)|$
- La relaxation continue, ou relaxation linéaire, consiste à relaxer les contraintes de domaines des variables de décision.
- On appelle solution relaxée ou solution continue d'un PLNE, la solution optimale continue de sa relaxation continue

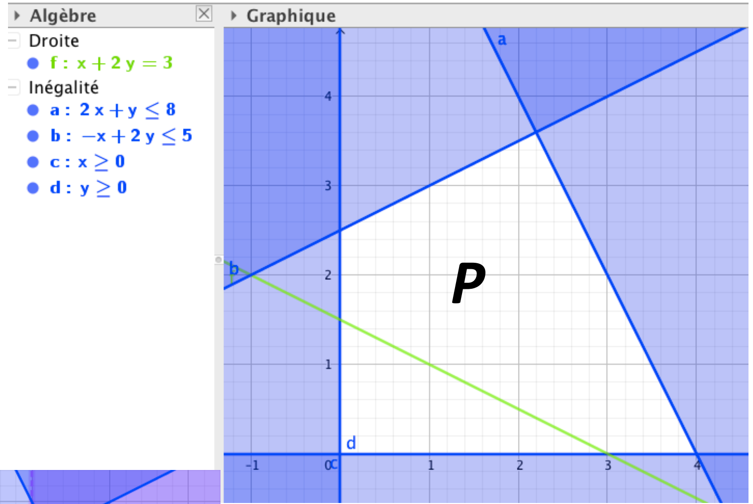
Gap de l'arrondi de la solution continue



- Solution arrondie peut être non réalisable
- Solution arrondie peut être très éloignée de la solution optimale entière :
  saut de relaxation très important.
- Arrondir peut être un non sens (cas des variables binaires)

La recherche arborescente (de la solution entière)

$$\begin{cases} Max \quad z = x + 2\,y \\ \qquad 2x + y \le 8 \\ \qquad -x + 2y \le 5 \\ \qquad x, y \in \mathbb{N} \end{cases}$$

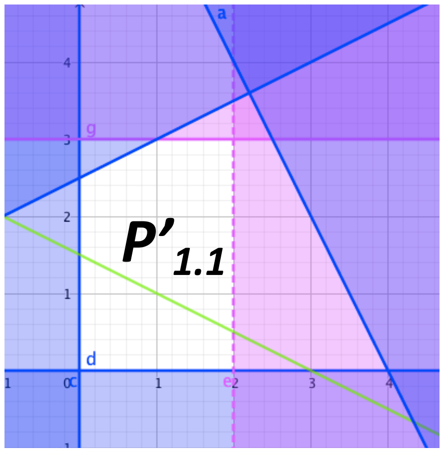


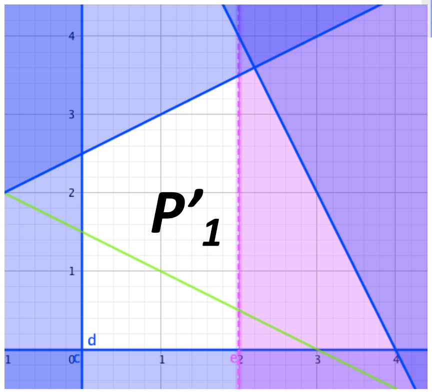
Solution relaxée :

z=9,4

x = 2,2

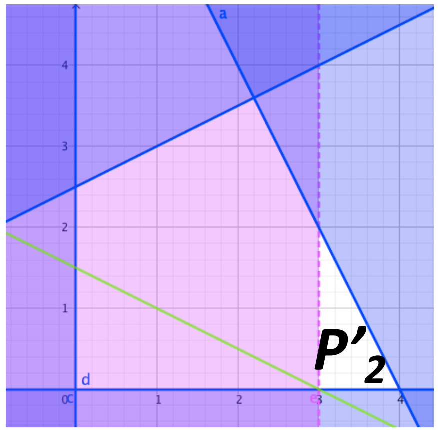
y = 3,6





x ≤ 2

x ≥ 3



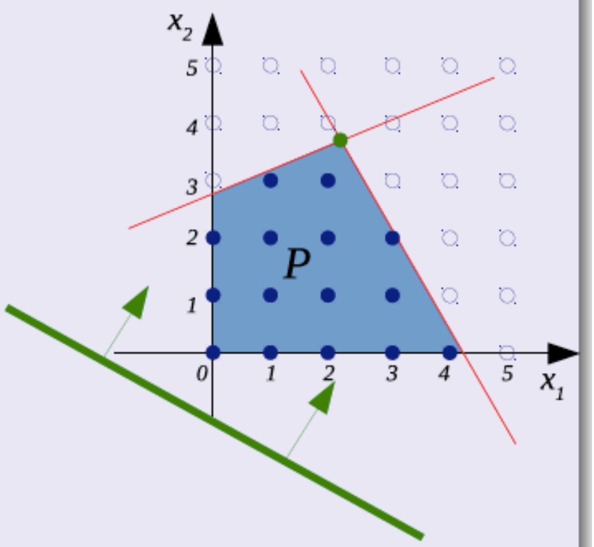
Séparation & Evaluation

Principe de la méthode (en maximisation)

- Choisir une variable x_i
- **Séparer** le problème en deux sous problèmes selon les valeur de x_i
  - en *PLNE*, choisir p tel que les deux sous problèmes correspondent à $x_i \leqslant p$ et $x_i \geqslant p+1$
  - en *PL* 0-1, les deux sous-problèmes correspondent à $x_i = 0$ et $x_i = 1$

  $\Rightarrow$ recherche arborescente : chaque nœud est un sous problème
- Supprimer un sous-problème P_s grâce à une **évaluation** de P_s :
  1. Soit $\overline{z_{P_s}}$, une borne supérieure du coût des solutions de P_s (en maximisation); par exemple, solution optimale du problème P_s relaxé obtenue par le Simplexe.
  2. Soit $\tilde{x}$, la meilleure solution entière de P connue, dont le coût est $\tilde{z}$

  $\Rightarrow$ Si $\overline{z_{P_s}} \leqslant \tilde{z}$, alors le sous-problème P_s n'est pas intéressant (il ne contient pas la solution optimale).



Méthode de Dakin

- Quelle évaluation ?
  - La relaxation continue du sous problème (évaluation par excès)
- Quelle variable pour brancher ?
  - La variable la plus proche d’un entier
- Quelle valeur (branche) choisir ?
  - Le sous problème le plus bas dans l’arbre (profondeur d’abord)
  - si plusieurs sous problème => celui avec la meilleure évaluation pour favoriser l’élagage !

Exemple

- $$\begin{cases} Max \quad 4x_1 + 3\,x_2 \\ \qquad\quad 3x_1 + 4x_2 \leq 24 \\ \qquad\quad 4x_1 + 2x_2 \leq 18 \\ \qquad\qquad x_1, x_2 \in \mathbb{N} \end{cases}$$

Efficacité

- Schéma classique
  - 10% des temps pour trouver la solution optimale
  - 90% du temps pour prouver qu'elle est optimale

  ➔ on peut stopper la méthode avant la fin (on perd la garanti d'optimalité)
- Limite des méthodes
  - PL : grands, plus de 100 000 variables
  - PL 0-1 : pas trop grand, quelques centaines de variables
  - PLNE : petits, quelques dizaines de variables

  ➔ heuristiques