

# Programmation linéaire

Nadia Brauner  
Julien Moncel  
Hervé Hocquard

2017-2018

# Plan

- 1 Introduction à la programmation linéaire
- 2 Interprétation géométrique
- 3 Bases et points extrêmes
- 4 L'algorithme du simplexe

# Programmation linéaire

## Cadre de la PL

### Programmation linéaire

nombre fini de variables réelles, contraintes linéaires, objectif linéaire

# Programmation linéaire

## Cadre de la PL

### Programmation linéaire

nombre fini de variables réelles, contraintes linéaires, objectif linéaire

Variables  $x_1, x_2, \dots, x_n$  réelles

# Programmation linéaire

## Cadre de la PL

### Programmation linéaire

nombre fini de variables réelles, contraintes linéaires, objectif linéaire

Variables  $x_1, x_2, \dots, x_n$  réelles

Contrainte générique (contrainte  $i$ ) :

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

# Programmation linéaire

## Cadre de la PL

### Programmation linéaire

nombre fini de variables réelles, contraintes linéaires, objectif linéaire

Variables  $x_1, x_2 \dots x_n$  réelles

Contrainte générique (contrainte  $i$ ) :

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

Fonction-objectif générique (à maximiser / minimiser) :

$$f(x_1, x_2 \dots x_n) = \sum_{j=1}^n c_j x_j$$

# Programmation linéaire

## Exemple : culture de courgettes et navets

Contraintes concernant les quantités d'engrais et d'anti-parasites

- $8\ell$  engrais A disponible  
→  $2\ell/m^2$  nécessaires pour courgettes,  $1\ell/m^2$  pour navets
- $7\ell$  engrais B disponible  
→  $1\ell/m^2$  nécessaires pour courgettes,  $2\ell/m^2$  pour navets
- $3\ell$  anti-parasites disponible  
→  $1\ell/m^2$  nécessaires pour navets

Objectif : produire le maximum (en poids) de légumes, sachant que rendements =  $4kg/m^2$  courgettes,  $5kg/m^2$  navets

# Programmation linéaire

## Exemple : culture de courgettes et navets

### Variables de décision

- $x_c$  : surface de courgettes
- $x_n$  : surface de navets



# Programmation linéaire

## Exemple : culture de courgettes et navets

### Variables de décision

- $x_c$  : surface de courgettes
- $x_n$  : surface de navets

**Fonction objectif**      $\max 4x_c + 5x_n$

# Programmation linéaire

## Exemple : culture de courgettes et navets

### Variables de décision

- $x_c$  : surface de courgettes
- $x_n$  : surface de navets

**Fonction objectif**      $\max 4x_c + 5x_n$

### Contraintes

- $2x_c + x_n \leq 8$      (engrais A)
- $x_c + 2x_n \leq 7$      (engrais B)
- $x_n \leq 3$      (anti-parasites)
- $x_c \geq 0$  et  $x_n \geq 0$

# Programmation linéaire

## **Intérêt de la PL**

Problème général d'optimisation sous contraintes

# Programmation linéaire

## Intérêt de la PL

Problème général d'optimisation sous contraintes

⇒ **AUCUNE** méthode **GÉNÉRALE** de résolution !!

# Programmation linéaire

## Intérêt de la PL

Problème général d'optimisation sous contraintes

⇒ **AUCUNE** méthode **GÉNÉRALE** de résolution !!

Problème linéaire quelconque

⇒ existence de méthodes de résolution générales et efficaces

# Programmation linéaire

## Intérêt de la PL

Problème général d'optimisation sous contraintes

⇒ **AUCUNE méthode GÉNÉRALE de résolution !!**

Problème linéaire quelconque

⇒ existence de méthodes de résolution générales et efficaces

Ces méthodes sont efficaces en théorie et en pratique

⇒ existence de nombreux logiciels de résolution :

Excel, CPLEX, Mathematica, LP-Solve...

# Programmation linéaire

## Intérêt de la PL

Problème général d'optimisation sous contraintes

⇒ **AUCUNE méthode GÉNÉRALE de résolution !!**

Problème linéaire quelconque

⇒ existence de méthodes de résolution générales et efficaces

Ces méthodes sont efficaces en théorie et en pratique

⇒ existence de nombreux logiciels de résolution :

Excel, CPLEX, Mathematica, LP-Solve...

## Cadre restrictif

- variables réelles
- contraintes linéaires
- objectif linéaire

# Programmation linéaire

## Représentation in extenso

- $\max 4x_c + 5x_n$
- $2x_c + x_n \leq 8$  (engrais A)
- $x_c + 2x_n \leq 7$  (engrais B)
- $x_n \leq 3$  (anti-parasites)
- $x_c \geq 0$  et  $x_n \geq 0$



# Programmation linéaire

## Représentation in extenso

- $\max 4x_c + 5x_n$
- $2x_c + x_n \leq 8$  (engrais A)
- $x_c + 2x_n \leq 7$  (engrais B)
- $x_n \leq 3$  (anti-parasites)
- $x_c \geq 0$  et  $x_n \geq 0$

## Représentation matricielle

$$\max (4 \ 5) \begin{pmatrix} x_c \\ x_n \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ x_n \end{pmatrix} \leq \begin{pmatrix} 8 \\ 7 \\ 3 \end{pmatrix}$$

$$x_c \geq 0 \quad x_n \geq 0$$

# Programmation linéaire

## Représentation in extenso

$$\max z = \sum_j c_j x_j$$

$$\text{s.c.} \quad \sum_j a_{ij} x_j \left\{ \begin{array}{l} \leq \\ \geq \\ = \end{array} \right\} b_i \quad i = 1, 2, \dots, m$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n$$

# Programmation linéaire

- second membre  $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$

- n var. de décision  $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$

- matrice de format  $m \times n$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \ddots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- coût (ou profit)  $c = (c_1, c_2 \dots c_n)$

# Programmation linéaire

- second membre  $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$

- matrice de format  $m \times n$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \ddots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- coût (ou profit)  $c = (c_1, c_2 \dots c_n)$

- n var. de décision  $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$

## Représentation matricielle

$$\max z = cX$$

$$\text{s.c.} \quad Ax \begin{cases} \leq \\ \geq \\ = \end{cases} b$$

$$x \geq 0$$

# Programmation linéaire

## Vocabulaire

- $x_i$  **variable** de décision du problème
- $x = (x_1, \dots, x_n)$  **solution réalisable** (admissible)  
*ssi* elle satisfait toutes les contraintes
- ensemble des solutions réalisables = **domaine** ou région admissible
- $x = (x_1, \dots, x_n)$  **solution optimale**  
*ssi* elle est réalisable et optimise la fonction-objectif

# Programmation linéaire

## Vocabulaire

- $x_i$  **variable** de décision du problème
- $x = (x_1, \dots, x_n)$  **solution réalisable** (admissible)  
ssi elle satisfait toutes les contraintes
- ensemble des solutions réalisables = **domaine** ou région admissible
- $x = (x_1, \dots, x_n)$  **solution optimale**  
ssi elle est réalisable et optimise la fonction-objectif
- **contraintes** inégalité ou égalité linéaire
  - $a_{11}x_1 + a_{12}x_2 \dots + a_{1n}x_n \leq b_1$
  - $a_{21}x_1 + a_{22}x_2 \dots + a_{2n}x_n \geq b_2$
  - $a_{31}x_1 + a_{32}x_2 \dots + a_{3n}x_n = b_3$

# Programmation linéaire

## Vocabulaire

- $x_i$  **variable** de décision du problème
- $x = (x_1, \dots, x_n)$  **solution réalisable** (admissible)  
ssi elle satisfait toutes les contraintes
- ensemble des solutions réalisables = **domaine** ou région admissible
- $x = (x_1, \dots, x_n)$  **solution optimale**  
ssi elle est réalisable et optimise la fonction-objectif
- **contraintes** inégalité ou égalité linéaire
  - $a_{11}x_1 + a_{12}x_2 \dots + a_{1n}x_n \leq b_1$
  - $a_{21}x_1 + a_{22}x_2 \dots + a_{2n}x_n \geq b_2$
  - $a_{31}x_1 + a_{32}x_2 \dots + a_{3n}x_n = b_3$
- **fonction objectif** (ou fonction économique) linéaire
  - $\max / \min c_1x_1 + c_2x_2 \dots + c_nx_n$

# Programmation linéaire

## Applications

### Feuille de TD : Programmation linéaire

- Production de vins
- Publicité
- Fabrication d'huile d'olives
- Bergamote

### Caseine :

- Cucumbers and onions
- Perfumes
- Dairy Products
- Apples



# Programmation linéaire

## Forme canonique d'un PL

- maximisation
- toutes les variables sont non négatives
- toutes les contraintes sont des inéquations du type " $\leq$ "

# Programmation linéaire

## Forme canonique d'un PL

- maximisation
- toutes les variables sont non négatives
- toutes les contraintes sont des inéquations du type " $\leq$ "

$$\max z = \sum_j c_j x_j$$

$$\text{s.c.} \quad \sum_j a_{ij} x_j \leq b_i \quad i = 1, 2, \dots, m$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n$$

- forme matricielle

$$\max z = cx$$

$$\text{s.c.} \quad Ax \leq b$$

$$x \geq 0$$

# Programmation linéaire

## Forme standard d'un PL

- maximisation
- toutes les variables sont non négatives
- toutes les contraintes sont des équations

# Programmation linéaire

## Forme standard d'un PL

- maximisation
- toutes les variables sont non négatives
- toutes les contraintes sont des équations

$$\max z = \sum_j c_j x_j$$

$$\text{s.c.} \quad \sum_j a_{ij} x_j = b_i \quad i = 1, 2, \dots, m$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n$$

- forme matricielle

$$\max z = cx$$

$$\text{s.c.} \quad Ax = b$$

$$x \geq 0$$

# Programmation linéaire

## Passage entre les formes

- équation  $\rightarrow$  inéquation

# Programmation linéaire

## Passage entre les formes

- équation  $\rightarrow$  inéquation

$$ax = b \iff \begin{cases} ax \leq b \\ ax \geq b \end{cases}$$

- max  $\leftrightarrow$  min

# Programmation linéaire

## Passage entre les formes

- équation  $\rightarrow$  inéquation

$$ax = b \iff \begin{cases} ax \leq b \\ ax \geq b \end{cases}$$

- $\max \leftrightarrow \min$        $\max f(x) = -\min -f(x)$
- inéquation  $\rightarrow$  équation :

# Programmation linéaire

## Passage entre les formes

- équation  $\rightarrow$  inéquation

$$ax = b \iff \begin{cases} ax \leq b \\ ax \geq b \end{cases}$$

- $\max \leftrightarrow \min$        $\max f(x) = -\min -f(x)$
- inéquation  $\rightarrow$  équation : ajouter une variable d'écart

$$\begin{aligned} ax \leq b &\iff ax + s = b, & s \geq 0 \\ ax \geq b &\iff ax - s = b, & s \geq 0 \end{aligned}$$

- variable non contrainte  $\rightarrow$  variables positives



# Programmation linéaire

## Passage entre les formes

- équation  $\rightarrow$  inéquation

$$ax = b \iff \begin{cases} ax \leq b \\ ax \geq b \end{cases}$$

- $\max \leftrightarrow \min$        $\max f(x) = -\min -f(x)$
- inéquation  $\rightarrow$  équation : ajouter une variable d'écart

$$\begin{aligned} ax \leq b &\iff ax + s = b, & s \geq 0 \\ ax \geq b &\iff ax - s = b, & s \geq 0 \end{aligned}$$

- variable non contrainte  $\rightarrow$  variables positives

$$x \leq 0 \iff \begin{cases} x = x^+ - x^- \\ x^+, x^- \geq 0 \end{cases}$$

# Programmation linéaire

## Passage entre les formes

Feuille de TD : Programmation linéaire

- Exercice Formes linéaires et canoniques

# Programmation linéaire

## Linéariser un problème non linéaire

$e_i$  : expression linéaire des variables de décision

- **obj** :  $\min \max\{e_1, e_2 \dots e_n\}$

# Programmation linéaire

## Linéariser un problème non linéaire

$e_i$  : expression linéaire des variables de décision

- **obj** :  $\min \max\{e_1, e_2 \dots e_n\}$

$$\begin{cases} \min y \\ y \geq e_i & i = 1, 2 \dots n \end{cases}$$

- **obj** :  $\max \min\{e_1, e_2 \dots e_n\}$

# Programmation linéaire

## Linéariser un problème non linéaire

$e_i$  : expression linéaire des variables de décision

- **obj** :  $\min \max\{e_1, e_2 \dots e_n\}$

$$\begin{cases} \min y \\ y \geq e_i & i = 1, 2 \dots n \end{cases}$$

- **obj** :  $\max \min\{e_1, e_2 \dots e_n\}$

$$\begin{cases} \max y \\ y \leq e_i & i = 1, 2 \dots n \end{cases}$$

- **obj** :  $\min |e_1|$

# Programmation linéaire

## Linéariser un problème non linéaire

$e_i$  : expression linéaire des variables de décision

- **obj** :  $\min \max\{e_1, e_2 \dots e_n\}$

$$\begin{cases} \min y \\ y \geq e_i \quad i = 1, 2 \dots n \end{cases}$$

- **obj** :  $\max \min\{e_1, e_2 \dots e_n\}$

$$\begin{cases} \max y \\ y \leq e_i \quad i = 1, 2 \dots n \end{cases}$$

- **obj** :  $\min |e_1|$

$$|e| = \max(e, -e) \quad \begin{cases} \min y \\ y \geq e_1 \\ y \geq -e_1 \end{cases} \quad \begin{cases} \min e^+ + e^- \\ e_1 = e^+ - e^- \\ e^+, e^- \geq 0 \end{cases}$$

# Programmation linéaire

## **Linéariser un problème non linéaire**

Feuille de TD : Programmation linéaire

- Exercice Linéarisation

# Programmation linéaire

## Un peu d'histoire

- années 30-40 : Kantorovitch, économiste soviétique  
⇒ modèles linéaires pour la planification et l'optimisation de la production
- années 40-50 : Dantzig, mathématicien américain  
⇒ algorithme du simplexe



# Programmation linéaire

## Un peu d'histoire

- années 30-40 : Kantorovitch, économiste soviétique  
⇒ modèles linéaires pour la planification et l'optimisation de la production
- années 40-50 : Dantzig, mathématicien américain  
⇒ algorithme du simplexe
- application historique
  - Opérations Vittles et Plainfare pour ravitaillement de la trizone pendant le blocus de Berlin par pont aérien (23 juin 1948 – 12 mai 1949)
  - simplexe exécuté à la main (des milliers de variables), jusqu'à 12 000 tonnes de matériel par jour !

# Programmation linéaire

## Un peu d'histoire

- années 30-40 : Kantorovitch, économiste soviétique  
⇒ modèles linéaires pour la planification et l'optimisation de la production
- années 40-50 : Dantzig, mathématicien américain  
⇒ algorithme du simplexe
- application historique
  - Opérations Vittles et Plainfare pour ravitaillement de la trizone pendant le blocus de Berlin par pont aérien (23 juin 1948 – 12 mai 1949)
  - simplexe exécuté à la main (des milliers de variables), jusqu'à 12 000 tonnes de matériel par jour !
- 1975 : prix Nobel économie Kantorovitch
- XXIème siècle : logiciels de PL disponibles partout, utilisation de la PL dans tous les domaines industriels...

# Interprétation géométrique

## Exemple : culture de courgettes et navets

### Variables de décision

- $x_c$  : surface de courgettes
- $x_n$  : surface de navets

**Fonction objectif**      $\max 4x_c + 5x_n$

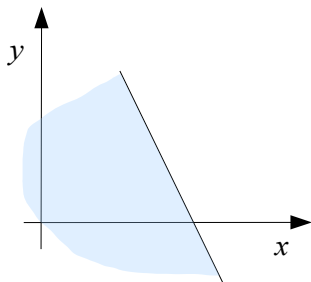
### Contraintes

- $2x_c + x_n \leq 8$  (engrais A)
- $x_c + 2x_n \leq 7$  (engrais B)
- $x_n \leq 3$  (anti-parasites)
- $x_c \geq 0$  et  $x_n \geq 0$

# Interprétation géométrique

**Interpréter les contraintes** courgettes et navets

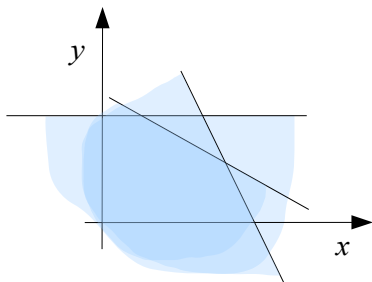
- $2x + y \leq 8 \Rightarrow$  demi-plan de  $\mathbb{R}^2$



# Interprétation géométrique

**Interpréter les contraintes** courgettes et navets

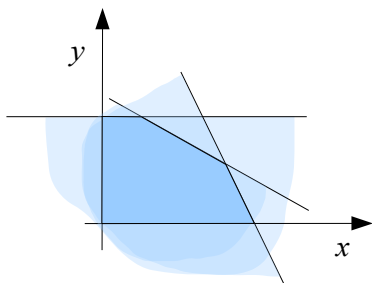
- $2x + y \leq 8 \Rightarrow$  demi-plan de  $\mathbb{R}^2$
- $x + 2y \leq 7 \Rightarrow$  demi-plan
- $y \leq 3 \Rightarrow$  demi-plan



# Interprétation géométrique

## Interpréter les contraintes courgettes et navets

- $2x + y \leq 8 \Rightarrow$  demi-plan de  $\mathbb{R}^2$
- $x + 2y \leq 7 \Rightarrow$  demi-plan
- $y \leq 3 \Rightarrow$  demi-plan
- $x \geq 0$  et  $y \geq 0 \Rightarrow$  demi-plans

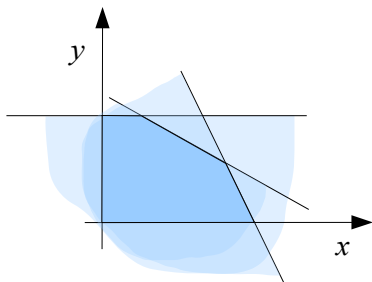


# Interprétation géométrique

**Interpréter les contraintes** courgettes et navets

- $2x + y \leq 8 \Rightarrow$  demi-plan de  $\mathbb{R}^2$
- $x + 2y \leq 7 \Rightarrow$  demi-plan
- $y \leq 3 \Rightarrow$  demi-plan
- $x \geq 0$  et  $y \geq 0 \Rightarrow$  demi-plans

Ensemble des solutions réalisables = intersection de ces demi-plans : **polyèdre**



# Interprétation géométrique

## Optimiser l'objectif

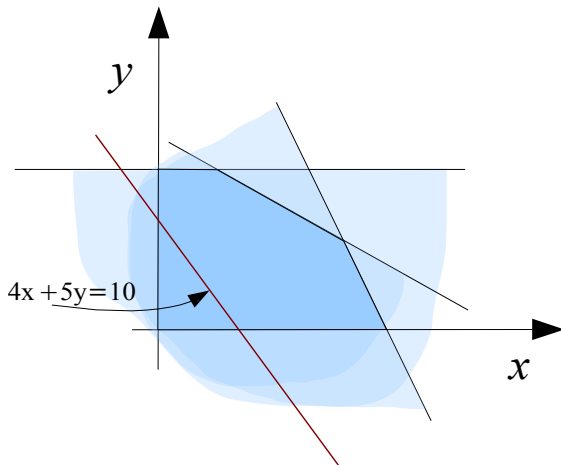
Les **lignes de niveau**  $\{4x + 5y = \text{constante}\}$  sont des droites parallèles



# Interprétation géométrique

## Optimiser l'objectif

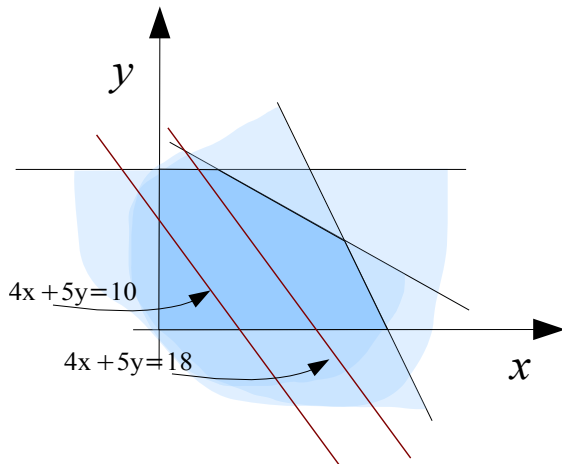
Les **lignes de niveau**  $\{4x + 5y = \text{constante}\}$  sont des droites parallèles



# Interprétation géométrique

## Optimiser l'objectif

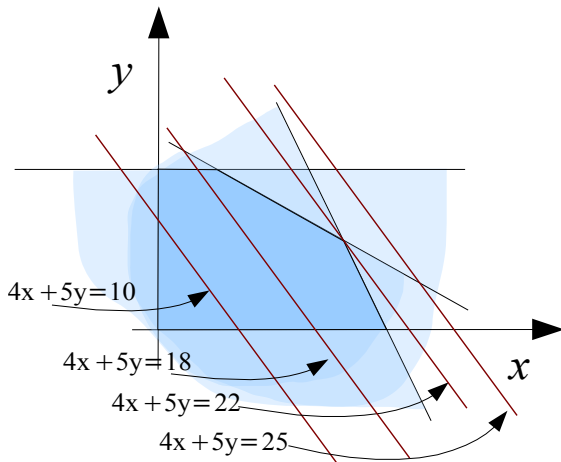
Les **lignes de niveau**  $\{4x + 5y = \text{constante}\}$  sont des droites parallèles



# Interprétation géométrique

## Optimiser l'objectif

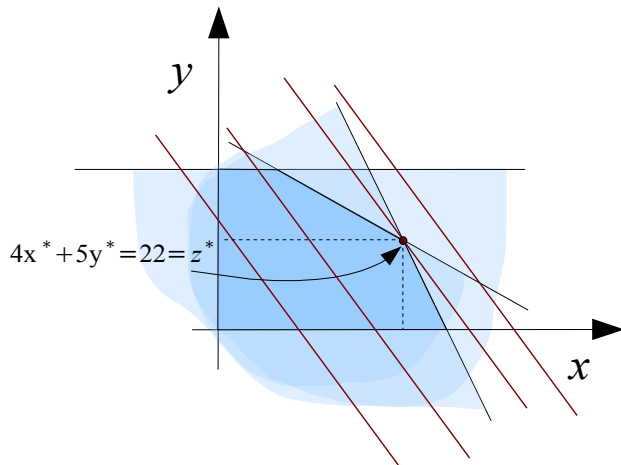
Les **lignes de niveau**  $\{4x + 5y = \text{constante}\}$  sont des droites parallèles



# Interprétation géométrique

## Optimiser l'objectif

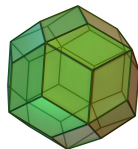
Les **lignes de niveau**  $\{4x + 5y = \text{constante}\}$  sont des droites parallèles



# Interprétation géométrique

## Géométrie d'un PL

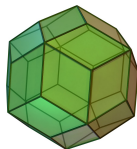
L'ensemble des solutions réalisables est toujours un **polyèdre** (intersection de demi-espaces)



# Interprétation géométrique

## Géométrie d'un PL

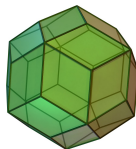
L'ensemble des solutions réalisables est toujours un **polyèdre** (intersection de demi-espaces)



Les lignes de niveau  $\{f = \text{constante}\}$  de la fonction-objectif  $f$  sont des **hyperplans affines** ( $n = 2 \Rightarrow$  droite,  $n = 3 \Rightarrow$  plan...)

# Interprétation géométrique

## Géométrie d'un PL



### Optimum atteint au bord

L'optimum de la fonction-objectif, s'il existe, est atteint en (au moins) un **sommet** du polyèdre.

Justification mathématique :

les dérivées partielles de  $f(x) = c \cdot x$  ne s'annulent jamais,

et le domaine  $\{x \mid \sum_{j=1}^n a_{ij}x_j \leq b_i, i = 1, \dots, m\}$  est compact

$\Rightarrow$  l'optimum est atteint au bord...

# Programmation linéaire

## Solutions d'un PL

La région admissible peut être

- vide
- non vide, bornée
- non vide, non bornée

Proposer des exemples de PL pour chacun des cas



# Programmation linéaire

## Solutions d'un PL

La région admissible peut être

- vide
  - nb solutions optimales : 0
- non vide, bornée
  - nb solutions optimales : 1 ou  $\infty$
- non vide, non bornée
  - nb solutions optimales : 0 ou 1 ou  $\infty$

Proposer des exemples de PL pour chacun des cas

# Programmation linéaire

## Solutions d'un PL

La région admissible peut être

- vide
  - nb solutions optimales : 0
- non vide, bornée
  - nb solutions optimales : 1 ou  $\infty$
- non vide, non bornée
  - nb solutions optimales : 0 ou 1 ou  $\infty$

Proposer des exemples de PL pour chacun des cas

Feuille de TD : Programmation linéaire

- Exercice Résolution graphique

# Bases et points extrêmes

## Rappels

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

- $A$  matrice  $m \times n$
- $x = (x_1 \ x_2 \ \dots \ x_n)$
- $b = (b_1 \ b_2 \ \dots \ b_m)$
- $c = (c_1 \ c_2 \ \dots \ c_n)$

- Les contraintes définissent un polyèdre
- Une solution optimale est un sommet du polyèdre

Comment énumérer les sommets d'un polyèdre ?

# Bases et points extrêmes

## Passage à la forme standard

### Forme standard

On peut rajouter des **variables d'écart** :

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

PL standard :

$$\begin{array}{ll} \max & z(x) = c \cdot x \\ \text{s.c} & Ax = b \\ & x \geq 0 \end{array}$$

# Bases et points extrêmes

## Passage à la forme standard

### Forme standard

On peut rajouter des **variables d'écart** :

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

PL standard :

$$\begin{array}{ll} \max & z(x) = c \cdot x \\ \text{s.c} & Ax = b \\ & x \geq 0 \end{array}$$

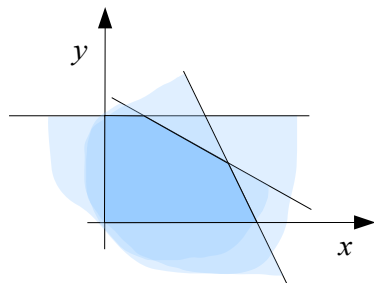
On travaille dans un espace de dimension plus grande, mais toutes les contraintes sont des égalités.

► Manipulations algébriques plus aisées

# Bases et points extrêmes

## Passage à la forme standard

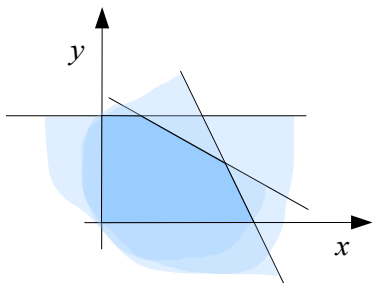
$$\begin{aligned} \max z &= 4x + 5y \\ \text{s.c. } 2x + y &\leq 8 \\ x + 2y &\leq 7 \\ y &\leq 3 \\ x, y &\geq 0 \end{aligned}$$



# Bases et points extrêmes

## Passage à la forme standard

$$\begin{aligned} \max z &= 4x + 5y \\ \text{s.c. } 2x + y &\leq 8 \\ x + 2y &\leq 7 \\ y &\leq 3 \\ x, y &\geq 0 \end{aligned}$$



$$\begin{aligned} \max z &= 4x + 5y \\ \text{s.c. } 2x + y + e_1 &= 8 \\ x + 2y + e_2 &= 7 \\ y + e_3 &= 3 \\ x, y, e_1, e_2, e_3 &\geq 0 \end{aligned}$$

9 points intéressants  
(intersection de contraintes)

5 points admissibles

énumération de ces 9 points  
comme solution de la forme  
standard (solutions de base)

# Bases et points extrêmes

$$\begin{array}{rcccccccc} \text{s.c.} & 2x & + & y & + & e_1 & & & = & 8 \\ & x & + & 2y & & & + & e_2 & = & 7 \\ & & & y & & & & & + & e_3 = & 3 \\ & x, & & y, & & e_1, & & e_2, & & e_3 & \geq & 0 \end{array}$$



# Bases et points extrêmes

$$\begin{array}{rclclclcl}
 \text{s.c.} & 2x & + & y & + & e_1 & & = & 8 \\
 & x & + & 2y & & & + & e_2 & = & 7 \\
 & & & y & & & & + & e_3 & = & 3 \\
 & x, & & y, & & e_1, & & e_2, & & e_3 & \geq & 0
 \end{array}$$

---

x	y	e <sub>1</sub>	e <sub>2</sub>	e <sub>3</sub>	sol de base	admiss.	pt extrême
---	---	----------------	----------------	----------------	-------------	---------	------------

---

# Bases et points extrêmes

$$\begin{array}{rclclclcl}
 \text{s.c.} & 2x & + & y & + & e_1 & & = & 8 \\
 & x & + & 2y & & & + & e_2 & = & 7 \\
 & & & y & & & & + & e_3 & = & 3 \\
 & x, & & y, & & e_1, & & e_2, & & e_3 & \geq & 0
 \end{array}$$

x	y	$e_1$	$e_2$	$e_3$	sol de base	admiss.	pt extrême
<u>0</u>	<u>0</u>						

## Bases et points extrêmes

$$\begin{array}{rclclclcl}
 \text{s.c.} & 2x & + & y & + & e_1 & & = & 8 \\
 & x & + & 2y & & & + & e_2 & = & 7 \\
 & & & y & & & & + & e_3 & = & 3 \\
 & x, & & y, & & e_1, & & e_2, & & e_3 & \geq & 0
 \end{array}$$

x	y	$e_1$	$e_2$	$e_3$	sol de base	admiss.	pt extrême
<u>0</u>	<u>0</u>	8	7	3	✓	✓	(0,0)

## Bases et points extrêmes

$$\begin{array}{rclclclcl}
 \text{s.c.} & 2x & + & y & + & e_1 & & = & 8 \\
 & x & + & 2y & & & + & e_2 & = & 7 \\
 & & & y & & & & + & e_3 & = & 3 \\
 & x, & & y, & & e_1, & & e_2, & & e_3 & \geq & 0
 \end{array}$$

x	y	e <sub>1</sub>	e <sub>2</sub>	e <sub>3</sub>	sol de base	admiss.	pt extrême
<u>0</u>	<u>0</u>	8	7	3	✓	✓	(0,0)
<u>0</u>	8	<u>0</u>	-9	-5	✓	✗	
<u>0</u>	3.5	4.5	<u>0</u>	-0.5	✓	✗	
<u>0</u>	3	5	1	<u>0</u>	✓	✓	(0,3)
4	<u>0</u>	<u>0</u>	3	3	✓	✓	(4,0)
7	<u>0</u>	-6	<u>0</u>	3	✓	✗	
	<u>0</u>			<u>0</u>	✗	✗	
3	2	<u>0</u>	<u>0</u>	1	✓	✓	(3,2)
2.5	3	<u>0</u>	-1.5	<u>0</u>	✓	✗	
1	3	3	<u>0</u>	<u>0</u>	✓	✓	(1,3)

## Bases et points extrêmes

$$\begin{array}{rclclclcl}
 \text{s.c.} & 2x & + & y & + & e_1 & & = & 8 \\
 & x & + & 2y & & & + & e_2 & = & 7 \\
 & & & y & & & & + & e_3 & = & 3 \\
 & x, & & y, & & e_1, & & e_2, & & e_3 & \geq & 0
 \end{array}$$

x	y	e <sub>1</sub>	e <sub>2</sub>	e <sub>3</sub>	sol de base	admiss.	pt extrême
<u>0</u>	<u>0</u>	8	7	3	✓	✓	(0,0)
<u>0</u>	8	<u>0</u>	-9	-5	✓	✗	
<u>0</u>	3.5	4.5	<u>0</u>	-0.5	✓	✗	
<u>0</u>	3	5	1	<u>0</u>	✓	✓	(0,3)
4	<u>0</u>	<u>0</u>	3	3	✓	✓	(4,0)
7	<u>0</u>	-6	<u>0</u>	3	✓	✗	
	<u>0</u>			<u>0</u>	✗	✗	
3	2	<u>0</u>	<u>0</u>	1	✓	✓	(3,2)
2.5	3	<u>0</u>	-1.5	<u>0</u>	✓	✗	
1	3	3	<u>0</u>	<u>0</u>	✓	✓	(1,3)

{points extrêmes}  $\iff$  {solutions de base admissibles}

# Bases et points extrêmes

- Système linéaire  $Ax = b$
- $A$  format  $m \times n$ , rang  $A = m \leq n$
- **Base** de  $A$  : sous-matrice  $B(m \times m)$  inversible de  $A$   
 $A = (B, N)$

$$(B, N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \quad \text{ou} \quad Bx_B + Nx_N = b$$

$$\Rightarrow x_B = B^{-1}b - B^{-1}Nx_N$$

- **Solution de base** associée à  $B$  :
  - $x_N = 0$  variables hors base
  - $x_B = B^{-1}b$  variables de base

# Bases et points extrêmes

## Applications

Feuille de TD : Programmation linéaire

- Exercice Bases \*2

# Bases et points extrêmes

## Base et solution de base

$$\begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \\ x, y, e_1, e_2, e_3 \geq 0 \end{cases}$$

Base initiale ?  $\{e_1, e_2, e_3\}$  par exemple :



# Bases et points extrêmes

## Base et solution de base

$$\begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \\ x, y, e_1, e_2, e_3 \geq 0 \end{cases}$$

Base initiale?  $\{e_1, e_2, e_3\}$  par exemple :

$$\begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \end{cases} \Leftrightarrow \begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases}$$

$e_1, e_2, e_3$  = variables de base,  $x, y$  = variables hors base

# Bases et points extrêmes

## Base et solution de base

$$\begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases}$$

- ▶ on met les variables hors base à 0
- ▶ on en déduit les valeur des variables de base

# Bases et points extrêmes

## Base et solution de base

$$\begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases}$$

- ▶ on met les variables hors base à 0
- ▶ on en déduit les valeur des variables de base

$$x = y = 0 \Rightarrow \begin{cases} e_1 = 8 - 2x - y = 8 \\ e_2 = 7 - x - 2y = 7 \\ e_3 = 3 - y = 3 \end{cases}$$

# Bases et points extrêmes

- $Ax = b, \quad x \geq 0$
- $(x_B, 0)$  associée à  $B$  est une **solution de base admissible** si

# Bases et points extrêmes

- $Ax = b, \quad x \geq 0$
- $(x_B, 0)$  associée à  $B$  est une **solution de base admissible** si  $x_B \geq 0$

# Bases et points extrêmes

- $Ax = b, \quad x \geq 0$
- $(x_B, 0)$  associée à  $B$  est une **solution de base admissible** si  $x_B \geq 0$
- **{points extrêmes du polyèdre}**  $\iff$  **{solutions de base admissibles du système linéaire correspondant}**
- nombre de points extrêmes  $\approx$

# Bases et points extrêmes

- $Ax = b, \quad x \geq 0$
- $(x_B, 0)$  associée à  $B$  est une **solution de base admissible** si  $x_B \geq 0$
- **{points extrêmes du polyèdre}**  $\iff$  **{solutions de base admissibles du système linéaire correspondant}**
- nombre de points extrêmes  $\approx C_n^m = \frac{n!}{m!(n-m)!}$
- solution de base dégénérée : certaines variables de base sont nulles
- si  $A$  est inversible : solution de base unique

# Bases et points extrêmes

## Base voisine et pivotage

### Bases voisines

Deux sommets voisins correspondent à deux bases  $B$  et  $B'$  telles qu'on remplace une variable de  $B$  pour obtenir  $B'$



# Bases et points extrêmes

## Base voisine et pivotage

### Bases voisines

Deux sommets voisins correspondent à deux bases  $B$  et  $B'$  telles qu'on remplace une variable de  $B$  pour obtenir  $B'$

► passer à un sommet voisin = changer de base (base voisine)

principe du pivotage

# Bases et points extrêmes

**Qui faire entrer dans la base ?**

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

# Bases et points extrêmes

## Qui faire entrer dans la base ?

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$

# Bases et points extrêmes

## Qui faire entrer dans la base ?

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$

# Bases et points extrêmes

## Qui faire entrer dans la base ?

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

# Bases et points extrêmes

## Qui faire entrer dans la base ?

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

Bilan :  $y_{\max} = 3$ , pour  $y = y_{\max}$  on a  $e_1 = 5 - 2x$ ,  $e_2 = 1 - x$ , et  $e_3 = 0$

# Bases et points extrêmes

## Qui faire entrer dans la base ?

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

Bilan :  $y_{\max} = 3$ , pour  $y = y_{\max}$  on a  $e_1 = 5 - 2x$ ,  $e_2 = 1 - x$ , et  $e_3 = 0$

► candidat pour une nouvelle base :

$$\{e_1, e_2, e_3\} \cup \{y\} \setminus \{e_3\} = \{e_1, e_2, y\}$$

$$(x, y, e_1, e_2, e_3) = (0, 3, 5, 1, 0)$$

# L'algorithme du simplexe

## Vers un algorithme de résolution

► Méthode de résolution "naïve" : énumérer tous les sommets, calculer  $f$  sur ces points, prendre le sommet pour lequel  $f$  est optimisé :

- fonctionne : nombre fini de sommets
- limitation : ce nombre peut être très grand en général...

**L'algorithme du simplexe** (G. B. Dantzig 1947) Algorithme itératif permettant de résoudre un problème de programmation linéaire.



# L'algorithme du simplexe

## Principe d'amélioration locale

À partir d'un sommet, chercher un sommet voisin qui améliore l'objectif.

## Principe d'amélioration locale (maximisation) :

Soit  $x_0$  sommet non optimum. Alors il existe  $x$ , un sommet **voisin** de  $x_0$ , tel que  $f(x) > f(x_0)$ .

# L'algorithme du simplexe

## Principe d'amélioration locale

À partir d'un sommet, chercher un sommet voisin qui améliore l'objectif.

## Principe d'amélioration locale (maximisation) :

Soit  $x_0$  sommet non optimum. Alors il existe  $x$ , un sommet **voisin** de  $x_0$ , tel que  $f(x) > f(x_0)$ .

► Méthode de résolution : on part d'un sommet  $x_0$  quelconque, on passe à un sommet voisin pour lequel  $f$  augmente, et ainsi de suite.

# L'algorithme du simplexe

## Principe d'amélioration locale

À partir d'un sommet, chercher un sommet voisin qui améliore l'objectif.

## Principe d'amélioration locale (maximisation) :

Soit  $x_0$  sommet non optimum. Alors il existe  $x$ , un sommet **voisin** de  $x_0$ , tel que  $f(x) > f(x_0)$ .

► Méthode de résolution : on part d'un sommet  $x_0$  quelconque, on passe à un sommet voisin pour lequel  $f$  augmente, et ainsi de suite.

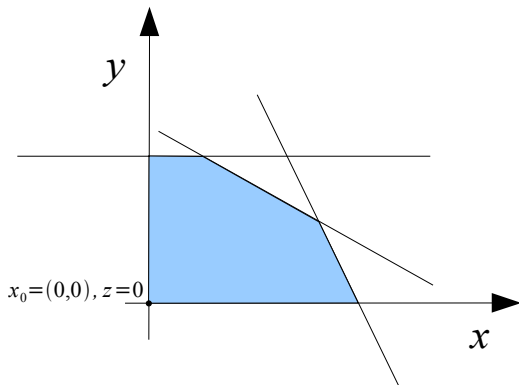
Remarque : on passe d'un problème **continu** (variables réelles) à un problème **discret** (nombre fini de sommets)...

# L'algorithme du simplexe

## Illustration 2D : courgettes et navets

$$x_0 = (0, 0), z = 0$$

$$z = 4x + 5y$$

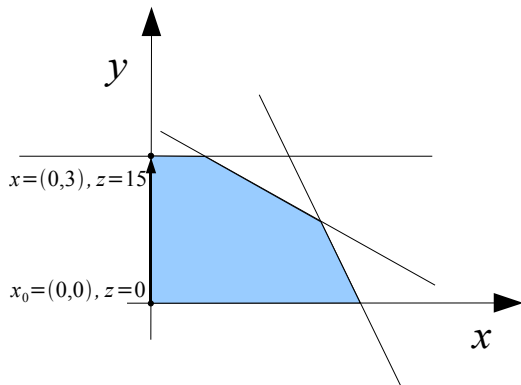


# L'algorithme du simplexe

## Illustration 2D : courgettes et navets

$x_0 = (0, 0), z = 0 \rightarrow x = (0, 3), z = 15$

$$z = 4x + 5y$$



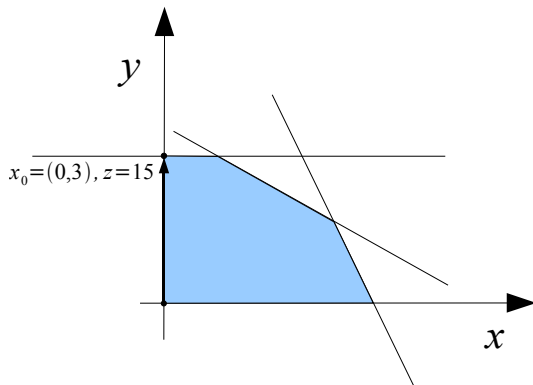
# L'algorithme du simplexe

## Illustration 2D : courgettes et navets

$$x_0 = (0, 0), z = 0 \rightarrow x = (0, 3), z = 15$$

$$x_0 = (0, 3), z = 15$$

$$z = 4x + 5y$$



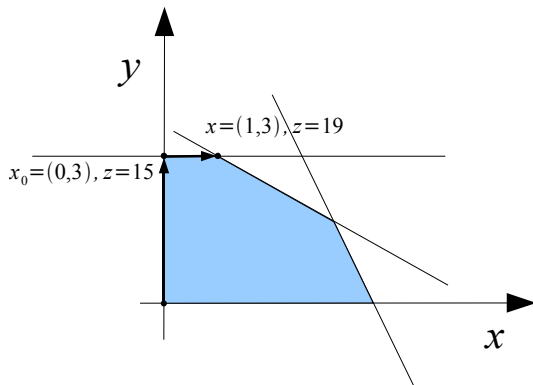
# L'algorithme du simplexe

## Illustration 2D : courgettes et navets

$$x_0 = (0, 0), z = 0 \rightarrow x = (0, 3), z = 15$$

$$x_0 = (0, 3), z = 15 \rightarrow x = (1, 3), z = 19$$

$$z = 4x + 5y$$



# L'algorithme du simplexe

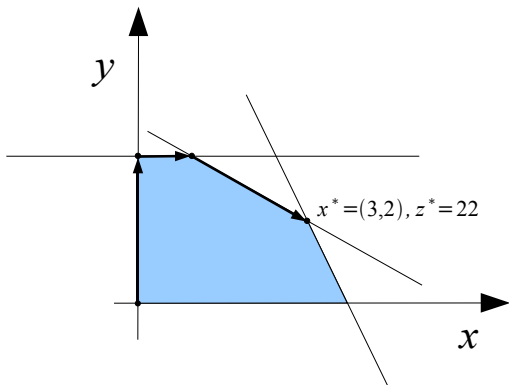
## Illustration 2D : courgettes et navets

$$x_0 = (0, 0), z = 0 \rightarrow x = (0, 3), z = 15$$

$$x_0 = (0, 3), z = 15 \rightarrow x = (1, 3), z = 19$$

$$x_0 = (1, 3), z = 19 \rightarrow x = (3, 2), z = 22$$

$$z = 4x + 5y$$



► plus d'amélioration locale possible  $\Rightarrow$  optimum



# L'algorithme du simplexe

## Illustration concrète

► Standardisation :

Maximiser  $z = 4x + 5y$

$$\text{s.c. } \begin{cases} 2x + y \leq 8 \\ x + 2y \leq 7 \\ y \leq 3 \\ x, y \geq 0 \end{cases}$$

# L'algorithme du simplexe

## Illustration concrète

► Standardisation :

$$\begin{array}{l} \text{Maximiser } z = 4x + 5y \\ \text{s.c. } \left\{ \begin{array}{l} 2x + y \leq 8 \\ x + 2y \leq 7 \\ y \leq 3 \\ x, y \geq 0 \end{array} \right. \end{array}$$

$$\begin{array}{l} \text{Maximiser } z = 4x + 5y \\ \text{s.c. } \left\{ \begin{array}{l} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \\ x, y, e_1, e_2, e_3 \geq 0 \end{array} \right. \end{array}$$

# L'algorithme du simplexe

## Illustration concrète

► Standardisation :

$$\begin{array}{l} \text{Maximiser } z = 4x + 5y \\ \text{s.c. } \left\{ \begin{array}{l} 2x + y \leq 8 \\ x + 2y \leq 7 \\ y \leq 3 \\ x, y \geq 0 \end{array} \right. \end{array}$$

$$\begin{array}{l} \text{Maximiser } z = 4x + 5y \\ \text{s.c. } \left\{ \begin{array}{l} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \\ x, y, e_1, e_2, e_3 \geq 0 \end{array} \right. \end{array}$$

Base initiale ?  $\{e_1, e_2, e_3\}$  par exemple :

# L'algorithme du simplexe

## Illustration concrète

► Standardisation :

$$\begin{array}{l} \text{Maximiser } z = 4x + 5y \\ \text{s.c. } \begin{cases} 2x + y \leq 8 \\ x + 2y \leq 7 \\ y \leq 3 \\ x, y \geq 0 \end{cases} \end{array}$$

$$\begin{array}{l} \text{Maximiser } z = 4x + 5y \\ \text{s.c. } \begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \\ x, y, e_1, e_2, e_3 \geq 0 \end{cases} \end{array}$$

Base initiale?  $\{e_1, e_2, e_3\}$  par exemple :

$$\begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \end{cases} \Leftrightarrow \begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases}$$

$e_1, e_2, e_3 =$  variables de base,  $x, y =$  variables hors base

# L'algorithme du simplexe

## **Solution de base associée**

- ▶ on met les variables hors base à 0
- ▶ on en déduit :
  - valeur des variables de base
  - valeur de  $z$

# L'algorithme du simplexe

## Solution de base associée

- ▶ on met les variables hors base à 0
- ▶ on en déduit :
  - valeur des variables de base
  - valeur de  $z$

$$\text{ici : } x = y = 0 \Rightarrow \begin{cases} e_1 = 8 - 2x - y = 8 \\ e_2 = 7 - x - 2y = 7 \\ e_3 = 3 - y = 3 \end{cases} \text{ et } z = 4x + 5y = 0$$

# L'algorithme du simplexe

## Changement de base

Observation essentielle :  $z = 4x + 5y = 0 \Rightarrow$  on peut augmenter  $z$  si  $x$  ou  $y$  rentre dans la base.

# L'algorithme du simplexe

## Changement de base

Observation essentielle :  $z = 4x + 5y = 0 \Rightarrow$  on peut augmenter  $z$  si  $x$  ou  $y$  rentre dans la base.

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?



# L'algorithme du simplexe

## Changement de base

Observation essentielle :  $z = 4x + 5y = 0 \Rightarrow$  on peut augmenter  $z$  si  $x$  ou  $y$  rentre dans la base.

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$

# L'algorithme du simplexe

## Changement de base

Observation essentielle :  $z = 4x + 5y = 0 \Rightarrow$  on peut augmenter  $z$  si  $x$  ou  $y$  rentre dans la base.

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$

# L'algorithme du simplexe

## Changement de base

Observation essentielle :  $z = 4x + 5y = 0 \Rightarrow$  on peut augmenter  $z$  si  $x$  ou  $y$  rentre dans la base.

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

# L'algorithme du simplexe

## Changement de base

Observation essentielle :  $z = 4x + 5y = 0 \Rightarrow$  on peut augmenter  $z$  si  $x$  ou  $y$  rentre dans la base.

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

Bilan :  $y_{\max} = 3$ , pour  $y = y_{\max}$  on a  $e_1 = 5 - x$ ,  $e_2 = 1 - x$ , et  $e_3 = 0$

# L'algorithme du simplexe

## Changement de base

Observation essentielle :  $z = 4x + 5y = 0 \Rightarrow$  on peut augmenter  $z$  si  $x$  ou  $y$  rentre dans la base.

Essayons avec  $y$  : quelle est la valeur max que pourra avoir  $y$  ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

Bilan :  $y_{\max} = 3$ , pour  $y = y_{\max}$  on a  $e_1 = 5 - x$ ,  $e_2 = 1 - x$ , et  $e_3 = 0$

► candidat pour une nouvelle base :

$$\{e_1, e_2, e_3\} \cup \{y\} \setminus \{e_3\} = \{e_1, e_2, y\}$$

# L'algorithme du simplexe

**Nouvelle base**  $\{e_1, e_2, y\}$

$$\begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases} \Rightarrow \begin{cases} e_1 = 8 - 2x - y = 5 - 2x + e_3 \\ e_2 = 7 - x - 2y = 1 - x + 2e_3 \\ y = 3 - e_3 \end{cases}$$

# L'algorithme du simplexe

**Nouvelle base**  $\{e_1, e_2, y\}$

$$\begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases} \Rightarrow \begin{cases} e_1 = 8 - 2x - y = 5 - 2x + e_3 \\ e_2 = 7 - x - 2y = 1 - x + 2e_3 \\ y = 3 - e_3 \end{cases}$$

Exprimons  $z$  en fonction des variables hors base

►  $z = 4x + 5y = 15 + 4x - 5e_3$

# L'algorithme du simplexe

**Nouvelle base**  $\{e_1, e_2, y\}$

$$\begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases} \Rightarrow \begin{cases} e_1 = 8 - 2x - y = 5 - 2x + e_3 \\ e_2 = 7 - x - 2y = 1 - x + 2e_3 \\ y = 3 - e_3 \end{cases}$$

Exprimons  $z$  en fonction des variables hors base

►  $z = 4x + 5y = 15 + 4x - 5e_3$

Solution de base associée

$$x = e_3 = 0 \Rightarrow \begin{cases} e_1 = 5 - 2x + e_3 = 5 \\ e_2 = 1 - x + 2e_3 = 1 \\ y = 3 - e_3 = 3 \end{cases} \quad \text{et} \quad z = 15$$



# L'algorithme du simplexe

## Itération

$z = 15 + 4x - 5e_3$  peut encore augmenter si  $x$  entre dans la base

# L'algorithme du simplexe

## Itération

$z = 15 + 4x - 5e_3$  peut encore augmenter si  $x$  entre dans la base

Si  $x$  entre, qui sort ?

Valeur max de  $x$  :

# L'algorithme du simplexe

## Itération

$z = 15 + 4x - 5e_3$  peut encore augmenter si  $x$  entre dans la base

Si  $x$  entre, qui sort ?

Valeur max de  $x$  :

- $e_1 = 5 - 2x + e_3 \geq 0 \Rightarrow x \leq 2.5$
- $e_2 = 1 - x + 2e_3 \geq 0 \Rightarrow x \leq 1$
- $y = 3 - e_3 \geq 0 \Rightarrow$  aucune contrainte sur  $x$

# L'algorithme du simplexe

## Itération

$z = 15 + 4x - 5e_3$  peut encore augmenter si  $x$  entre dans la base

Si  $x$  entre, qui sort ?

Valeur max de  $x$  :

- $e_1 = 5 - 2x + e_3 \geq 0 \Rightarrow x \leq 2.5$
- $e_2 = 1 - x + 2e_3 \geq 0 \Rightarrow x \leq 1$
- $y = 3 - e_3 \geq 0 \Rightarrow$  aucune contrainte sur  $x$

Bilan :  $x_{\max} = 1$  et  $e_2$  sort.

Nouvelle base  $\{e_1, y, x\}$

# L'algorithme du simplexe

## Itération

$z = 15 + 4x - 5e_3$  peut encore augmenter si  $x$  entre dans la base

Si  $x$  entre, qui sort ?

Valeur max de  $x$  :

- $e_1 = 5 - 2x + e_3 \geq 0 \Rightarrow x \leq 2.5$
- $e_2 = 1 - x + 2e_3 \geq 0 \Rightarrow x \leq 1$
- $y = 3 - e_3 \geq 0 \Rightarrow$  aucune contrainte sur  $x$

Bilan :  $x_{\max} = 1$  et  $e_2$  sort.

Nouvelle base  $\{e_1, y, x\}$

$$\begin{cases} e_1 = 3 + 2e_2 - 3e_3 \\ x = 1 - e_2 + 2e_3 \\ y = 3 - e_3 \\ z = 19 - 4e_2 + 3e_3 \end{cases}$$

# L'algorithme du simplexe

## Itération (suite)

$z = 19 - 4e_2 + 3e_3$  peut encore augmenter si  $e_3$  entre dans la base

# L'algorithme du simplexe

## Itération (suite)

$z = 19 - 4e_2 + 3e_3$  peut encore augmenter si  $e_3$  entre dans la base

Si  $e_3$  entre, qui sort ?

Valeur max de  $e_3$  :

# L'algorithme du simplexe

## Itération (suite)

$z = 19 - 4e_2 + 3e_3$  peut encore augmenter si  $e_3$  entre dans la base

Si  $e_3$  entre, qui sort ?

Valeur max de  $e_3$  :

- $e_1 = 3 + 2e_2 - 3e_3 \geq 0 \Rightarrow e_3 \leq 1$
- $x = 1 - e_2 + 2e_3 \geq 0 \Rightarrow$  aucune contrainte sur  $e_3$
- $y = 3 - e_3 \geq 0 \Rightarrow e_3 \leq 3$



# L'algorithme du simplexe

## Itération (suite)

$z = 19 - 4e_2 + 3e_3$  peut encore augmenter si  $e_3$  entre dans la base

Si  $e_3$  entre, qui sort ?

Valeur max de  $e_3$  :

- $e_1 = 3 + 2e_2 - 3e_3 \geq 0 \Rightarrow e_3 \leq 1$
- $x = 1 - e_2 + 2e_3 \geq 0 \Rightarrow$  aucune contrainte sur  $e_3$
- $y = 3 - e_3 \geq 0 \Rightarrow e_3 \leq 3$

Bilan :  $e_{3_{\max}} = 1$ ,  $e_1$  sort. Nouvelle base  $\{e_3, y, x\}$  :

# L'algorithme du simplexe

## Itération (suite)

$z = 19 - 4e_2 + 3e_3$  peut encore augmenter si  $e_3$  entre dans la base

Si  $e_3$  entre, qui sort ?

Valeur max de  $e_3$  :

- $e_1 = 3 + 2e_2 - 3e_3 \geq 0 \Rightarrow e_3 \leq 1$
- $x = 1 - e_2 + 2e_3 \geq 0 \Rightarrow$  aucune contrainte sur  $e_3$
- $y = 3 - e_3 \geq 0 \Rightarrow e_3 \leq 3$

Bilan :  $e_{3_{\max}} = 1$ ,  $e_1$  sort. Nouvelle base  $\{e_3, y, x\}$  :

$$\begin{cases} e_3 = 1 + 2/3e_2 - 1/3e_1 \\ x = 3 + 1/3e_2 - 2/3e_1 \\ y = 2 - 2/3e_2 + 1/3e_1 \\ z = 22 - 2e_2 - e_1 \end{cases}$$

# L'algorithme du simplexe

## Terminaison

On a  $z = 22 - 2e_2 - e_1$ , donc  $z^* \leq 22$

# L'algorithme du simplexe

## Terminaison

On a  $z = 22 - 2e_2 - e_1$ , donc  $z^* \leq 22$

Or la solution de base  $x = 3, y = 2, e_3 = 1$  donne  $z = 22$

► optimum

# L'algorithme du simplexe

## Terminaison

On a  $z = 22 - 2e_2 - e_1$ , donc  $z^* \leq 22$

Or la solution de base  $x = 3, y = 2, e_3 = 1$  donne  $z = 22$

► optimum

La condition de terminaison concerne les coefficients de  $z$  exprimée avec les variables hors base.

# L'algorithme du simplexe

## Coûts réduits

$B$ , une base de  $Ax = b$

la fonction objectif :

$$\begin{aligned}z &= cX = c_B x_B + c_N x_N \\ &= c_B B^{-1} b - (c_B B^{-1} N - c_N) x_N \\ &= z_0 - \sum_{j=1}^n (c_B B^{-1} a^j - c_j) x_j \\ &= z_0 - \sum_{j=1}^n (z_j - c_j) x_j\end{aligned}$$

$z_j - c_j = c_B B^{-1} a^j - c_j$  est le coût réduit de la variable hors base  $x_j$

# L'algorithme du simplexe

à chaque itération

	$x_N$	$x_B$
$z = z_0$	coûts réduits	0

$$x_B = \oplus \quad \dots$$

à l'optimum

	$x_N$	$x_B$
$z = z_0^*$	$\ominus$	0

$$x_B = \oplus \quad \dots$$

# L'algorithme du simplexe

Principe heuristique : faire rentrer en base la variable avec le coefficient le plus grand

Qui faire sortir ?

## Principe du quotient minimal

colonne pivot  $x_1$     second membre  $\geq 0$     quotient

$a_1 > 0$	$b_1$	-
$a_2 < 0$	$b_2$	$-\frac{b_2}{a_2}$
$a_3 < 0$	$b_3$	$-\frac{b_3}{a_3}$
$a_4 = 0$	$b_4$	-

ligne  $r$      $\frac{b_r}{a_r} = \min \left\{ -\frac{b_i}{a_i} \mid a_i < 0 \right\}$



# L'algorithme du simplexe

Principe heuristique : faire rentrer en base la variable avec le coefficient le plus grand

Qui faire sortir ?

## Principe du quotient minimal

colonne pivot  $x_1$     second membre  $\geq 0$     quotient

$a_1 > 0$	$b_1$	-
$a_2 < 0$	$b_2$	$-\frac{b_2}{a_2}$
$a_3 < 0$	$b_3$	$-\frac{b_3}{a_3}$
$a_4 = 0$	$b_4$	-

ligne  $r$      $\frac{b_r}{a_r} = \min \left\{ -\frac{b_i}{a_i} \mid a_i < 0 \right\}$

# L'algorithme du simplexe

## Pivotage

$$\begin{array}{rcccl}
 & c & \cdots & a & \\
 & \vdots & & \vdots & \\
 \text{ligne pivot} - & p & \cdots & b & \implies a \rightarrow a - \frac{b}{p}c \\
 & | & & & \\
 & \text{colonne} & & & \\
 & \text{pivot} & & & 
 \end{array}$$

# L'algorithme du simplexe

## Phase II

Données : un programme linéaire et une solution de base admissible

Résultat : une solution de base admissible optimale ou déclarer  
"PL non borné"

- 1 Choix d'une colonne (variable) entrante
  - choisir une variable hors base  $x_j$  (colonne) ayant un coût réduit négatif
  - s'il n'existe pas de colonne entrante : STOP, la solution de base est optimale
- 2 Choix d'une ligne (variable) sortante
  - Choisir une ligne  $r$  minimisant le quotient
  - s'il n'existe pas de ligne sortante : STOP le tableau courant est non borné
- 3 Mise à jour de la base et du tableau
  - pivoter autour de  $a_{rj}$  et retourner en (1)

# L'algorithme du simplexe

- Solution de base dégénérée si une ou plusieurs variables de base sont zéros (plus de bijection entre les solutions de base admissibles et les points extrêmes)
- Si toutes les solutions de base admissibles sont non dégénérées, l'algorithme du simplexe termine après un nombre fini d'itérations

# L'algorithme du simplexe

## Phase I

La Phase II de l'algorithme du simplexe prend en entrée une base réalisable. La Phase I permet de trouver une première base réalisable.

### Découvrez la Phase I

- **Caseine** : Phase I\*
- **Feuille d'exercice** : Simplexe \*2