Cours 4: Programmation linéaire

- Position du problème
- Algorithme du simplexe générique
- Dualité.
- Dégénérescence et terminaison de l'algorithme

Cours 4: Programmation linéaire

- Position du problème
- Algorithme du simplexe générique
- Dualité.
- Dégénérescence et terminaison de l'algorithme

Un problème de programmation linéaire

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Problème: Trouver un point $x = (x_1, \ldots, x_n)$ qui

- satisfait les m contraintes $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$,
- et maximise le produit $c \cdot x = c_1x_1 + \ldots + c_nx_n$.

Un problème de programmation linéaire

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Problème: Trouver un point $x = (x_1, \ldots, x_n)$ qui

- satisfait les m contraintes $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$,
- et maximise le produit $c \cdot x = c_1x_1 + \ldots + c_nx_n$.

Reformulation matricielle: on cherche $\max(c \cdot x \mid Ax \leq b)$.

Un problème de programmation linéaire

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Problème: Trouver un point $x = (x_1, \ldots, x_n)$ qui

- satisfait les m contraintes $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$,
- et maximise le produit $c \cdot x = c_1x_1 + \ldots + c_nx_n$.

Reformulation matricielle: on cherche $\max(c \cdot x \mid Ax \leq b)$.

Remarque: Les $a_{i,j}$, b_i et c_i peuvent être négatifs, on peut donc modéliser des contraintes $\geq$ ou $=$ et chercher $\min$ au lieu de $\max$.

Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.

Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros
- 10 lys, 20 roses, 10 jonquilles: 5 euros

Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.
Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros
- 10 lys, 20 roses, 10 jonquilles: 5 euros

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.
Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros x bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros y bouquets

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.

Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros — x bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros — y bouquets

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Contraintes: sur les lys: $10x + 10y \leq 50$

sur les roses: $10x + 20y \leq 80$

sur les jonquilles: $20x + 10y \leq 80$

Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.
Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros x bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros y bouquets

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Contraintes: sur les lys: $10x + 10y \leq 50$

sur les roses: $10x + 20y \leq 80$

sur les jonquilles: $20x + 10y \leq 80$

$$x \geq 0 \qquad y \geq 0$$

Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.

Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros $\quad x$ bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros $\quad y$ bouquets

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Contraintes: sur les lys: $10x + 10y \leq 50$

sur les roses: $10x + 20y \leq 80$

sur les jonquilles: $20x + 10y \leq 80$

$x \geq 0 \qquad y \geq 0$

Fonction économique à maximiser: $\max(4x + 5y)$.

Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.

Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros — x bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros — y bouquets

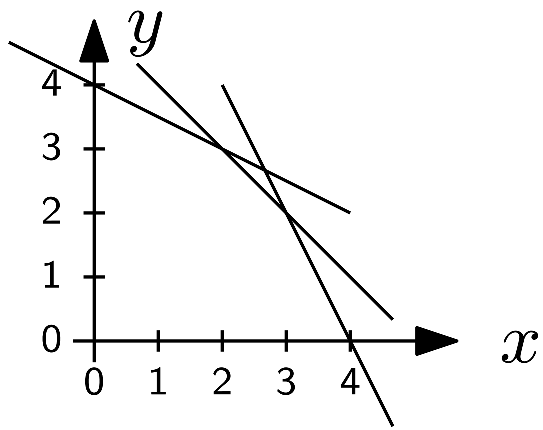
Problème Quels bouquets préparer si on est assuré de tout vendre ?

Contraintes: sur les lys: $10x + 10y \leq 50$

sur les roses: $10x + 20y \leq 80$

sur les jonquilles: $20x + 10y \leq 80$

$x \geq 0 \qquad y \geq 0$



Fonction économique à maximiser: $\max(4x + 5y)$.

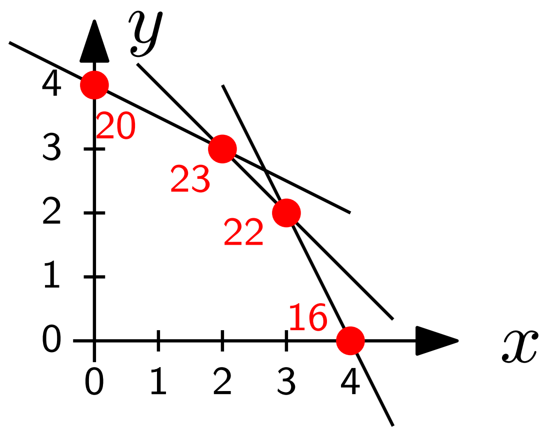
Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.
Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros x bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros y bouquets

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Contraintes: sur les lys: $10x + 10y \leq 50$
sur les roses: $10x + 20y \leq 80$
sur les jonquilles: $20x + 10y \leq 80$
$x \geq 0 \qquad y \geq 0$



Fonction économique à maximiser: $\max(4x + 5y)$.

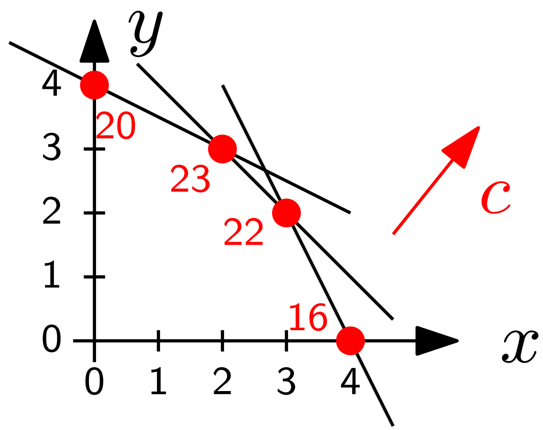
Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.
Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros — x bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros — y bouquets

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Contraintes: sur les lys: $10x + 10y \leq 50$
sur les roses: $10x + 20y \leq 80$
sur les jonquilles: $20x + 10y \leq 80$
$x \geq 0 \qquad y \geq 0$



Fonction économique à maximiser: $\max(4x + 5y)$.

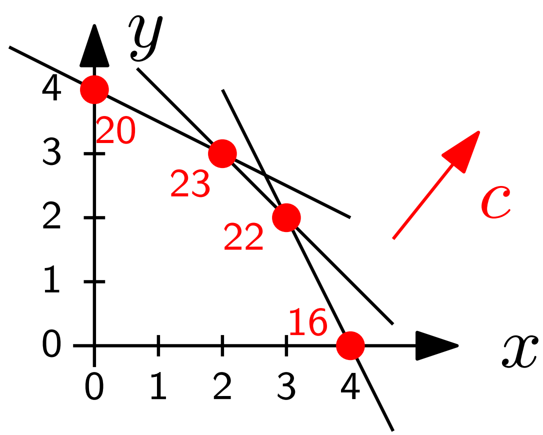
Un exemple bateau: le fleuriste

Donnée. En stock: 50 lys, 80 roses, 80 jonquilles.
Composition des bouquets au catalogue:

- 10 lys, 10 roses, 20 jonquilles: 4 euros — x bouquets
- 10 lys, 20 roses, 10 jonquilles: 5 euros — y bouquets

Problème Quels bouquets préparer si on est assuré de tout vendre ?

Contraintes: sur les lys: $10x + 10y \leq 50$
sur les roses: $10x + 20y \leq 80$
sur les jonquilles: $20x + 10y \leq 80$
$x \geq 0 \qquad y \geq 0$



Fonction économique à maximiser: $\max(4x + 5y)$.

Lorsqu'on a les contraintes $x_i \geq 0$ on peut toujours penser en termes économiques: produits finis, matières premières, bénéfice.

Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Interprétation géométrique

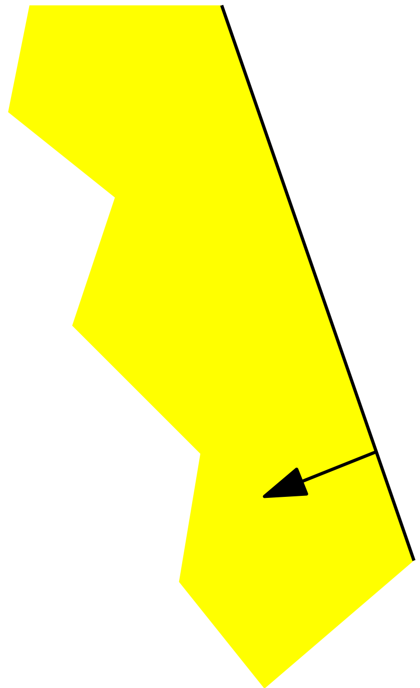
Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

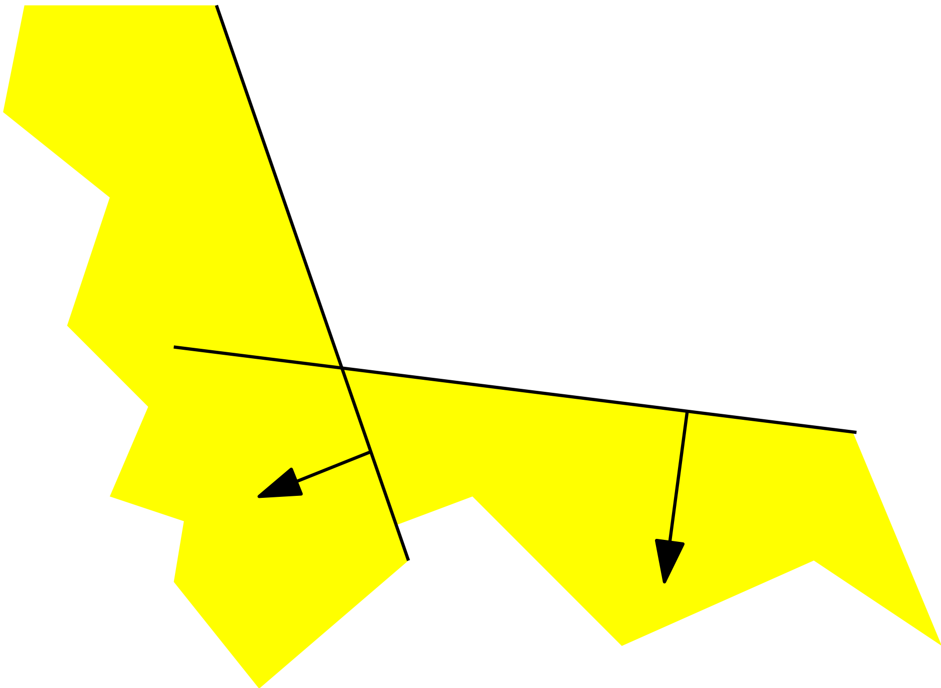
Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.



Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

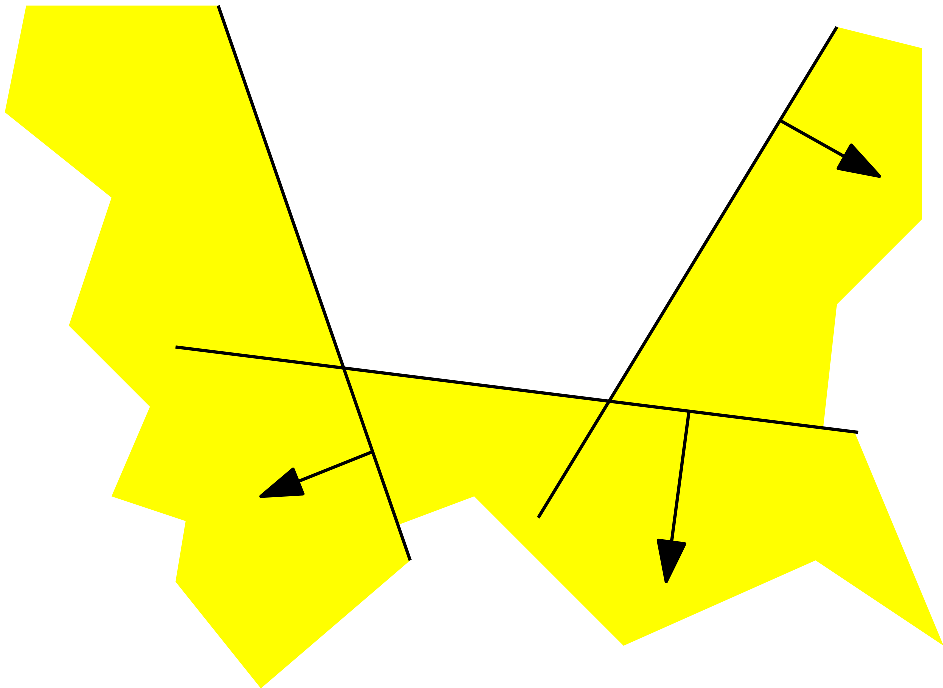
Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.



Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

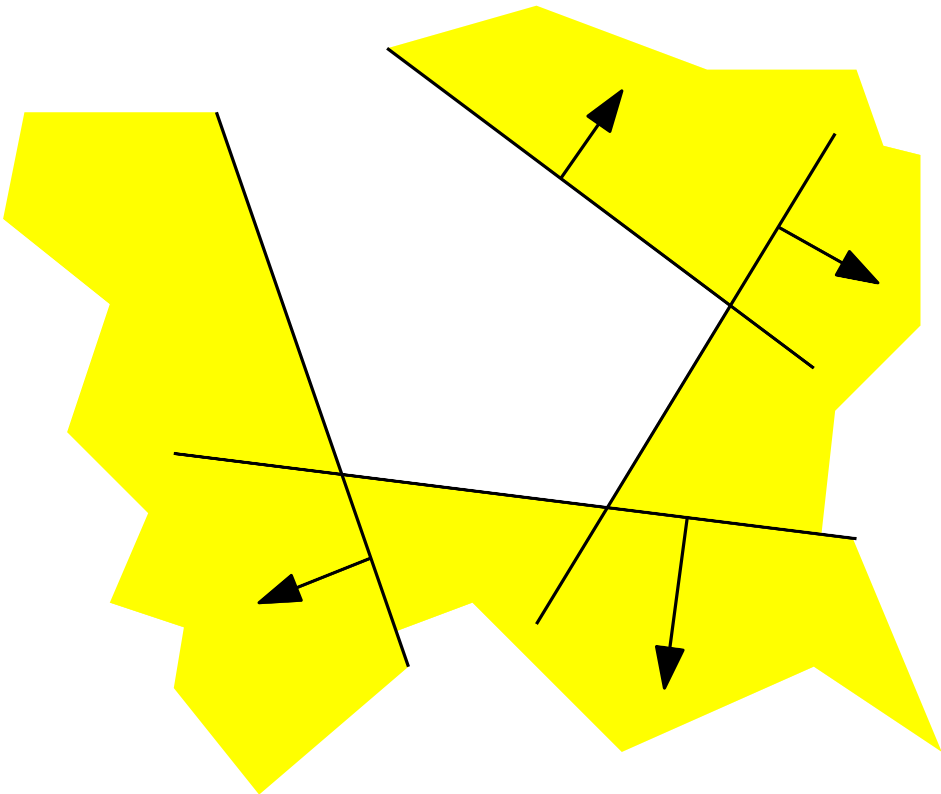
Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.



Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

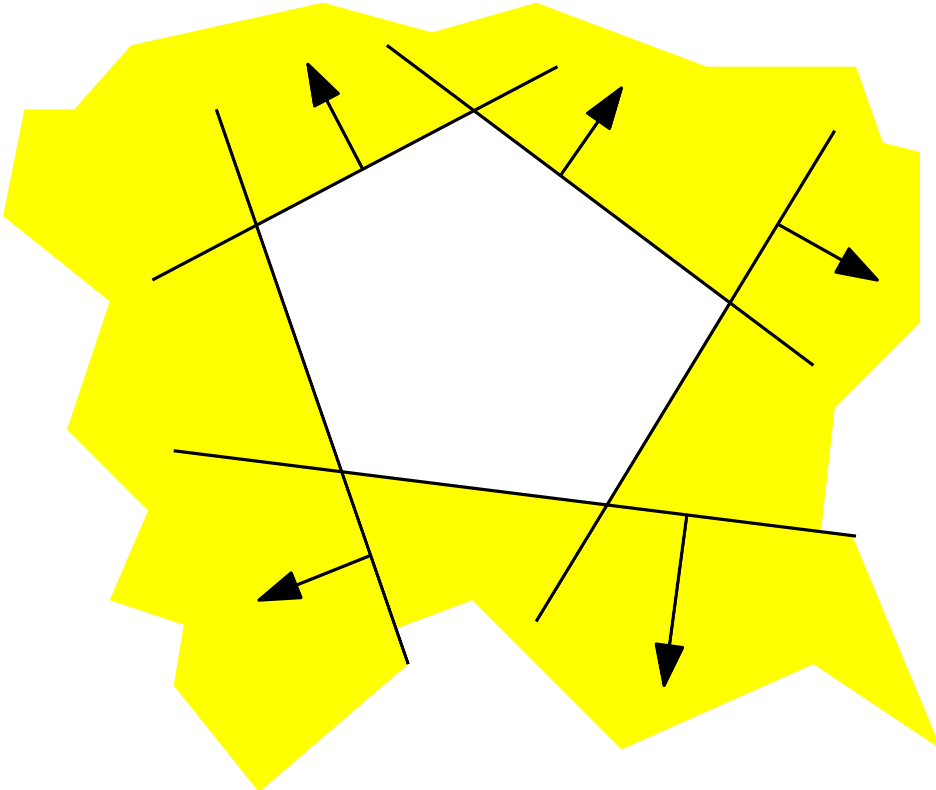
Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.



Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

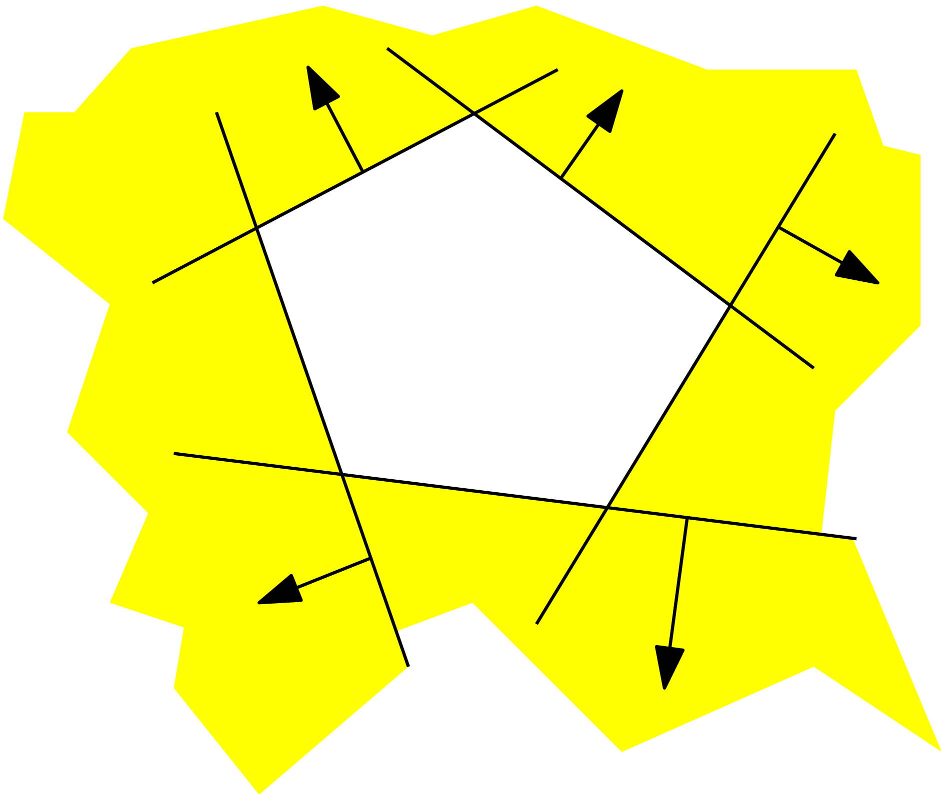


Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$ forme un polyhèdre convexe P

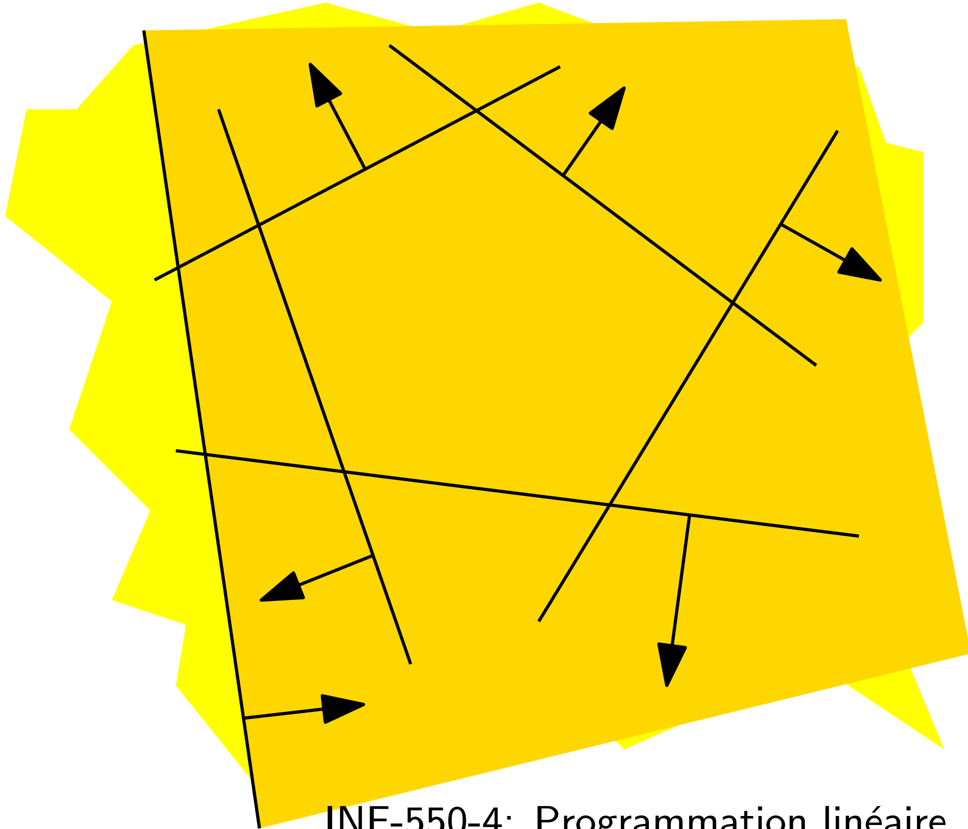


Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$ forme un polyhèdre convexe P éventuellement vide

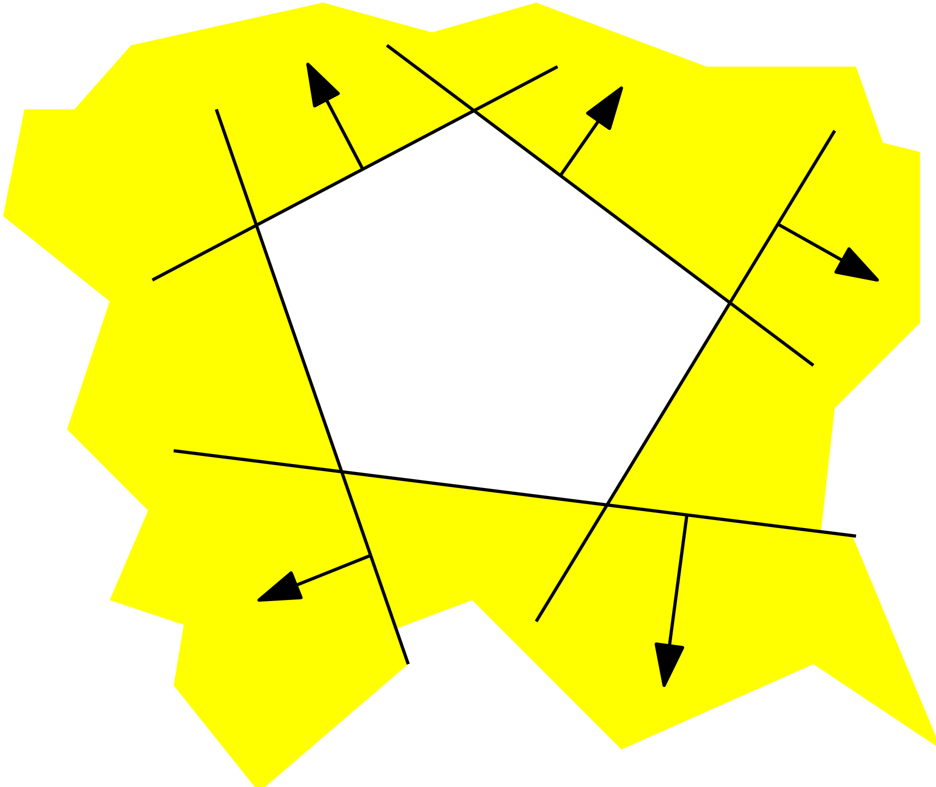


Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$ forme un polyhèdre convexe P éventuellement vide

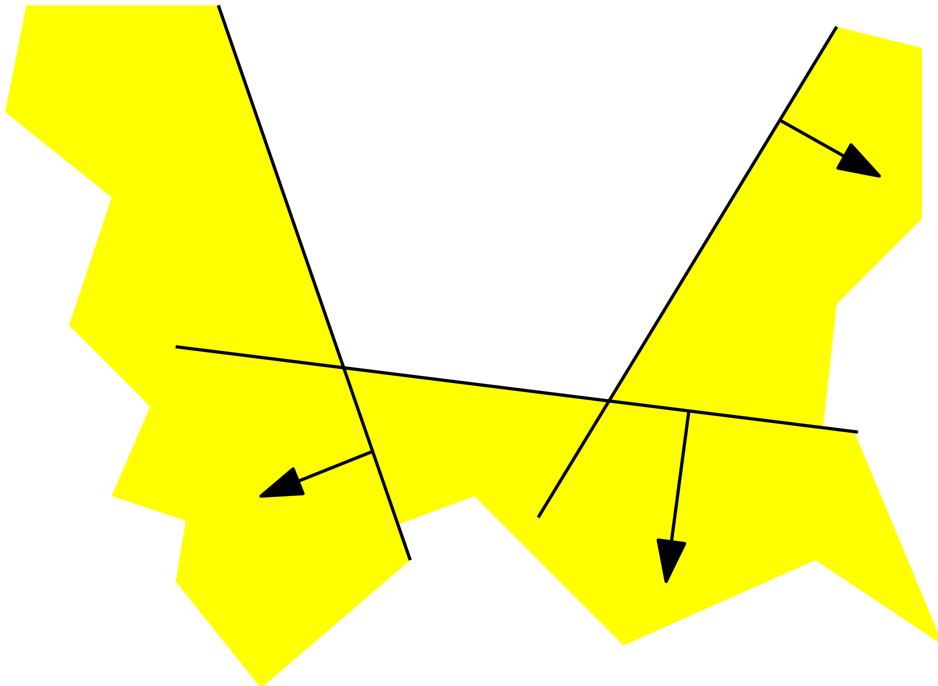


Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$ forme un polyhèdre convexe P éventuellement vide ou non borné



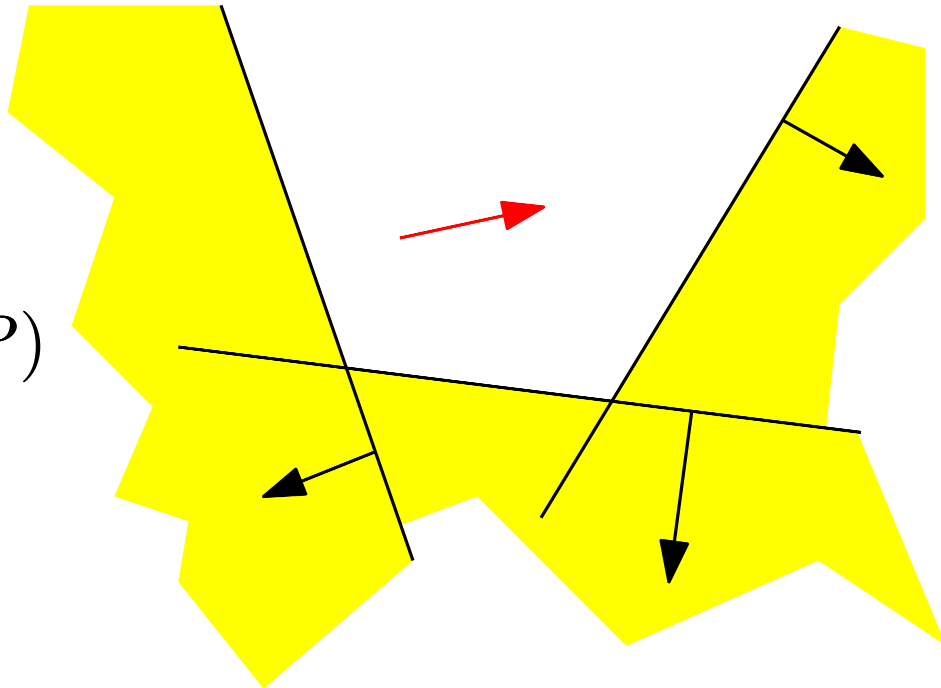
Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$ forme un polyhèdre convexe P éventuellement vide ou non borné

Le vecteur c indique la direction dans laquelle on optimise: $\max(c \cdot x \mid x \in P)$

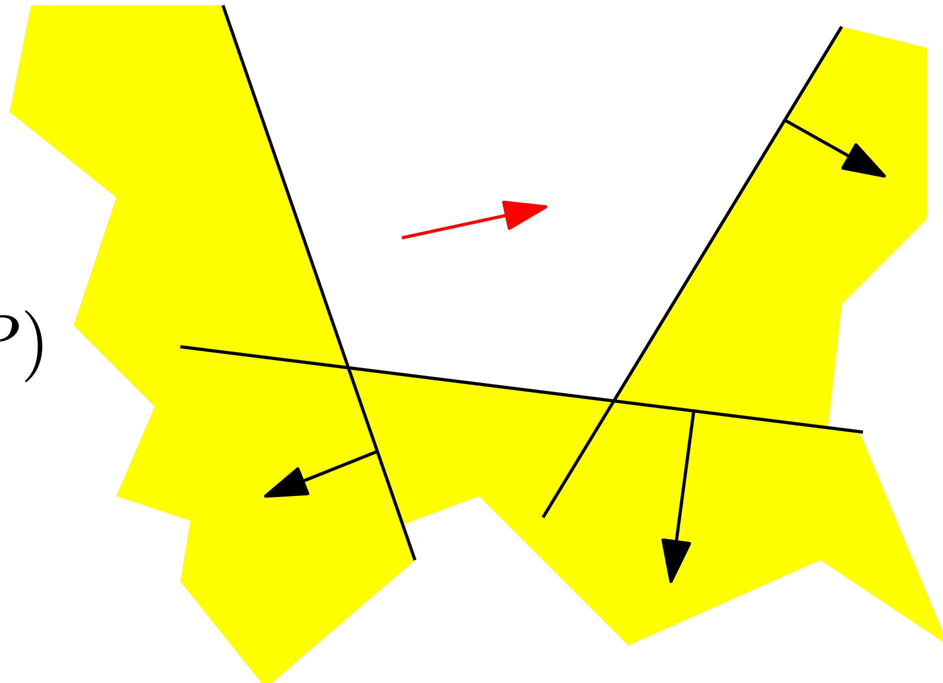


Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$
forme un polyhèdre convexe P
éventuellement vide ou non borné
Le vecteur c indique la direction dans laquelle on optimise: $\max(c \cdot x \mid x \in P)$
L'optimum peut être à l'infini.



Interprétation géométrique

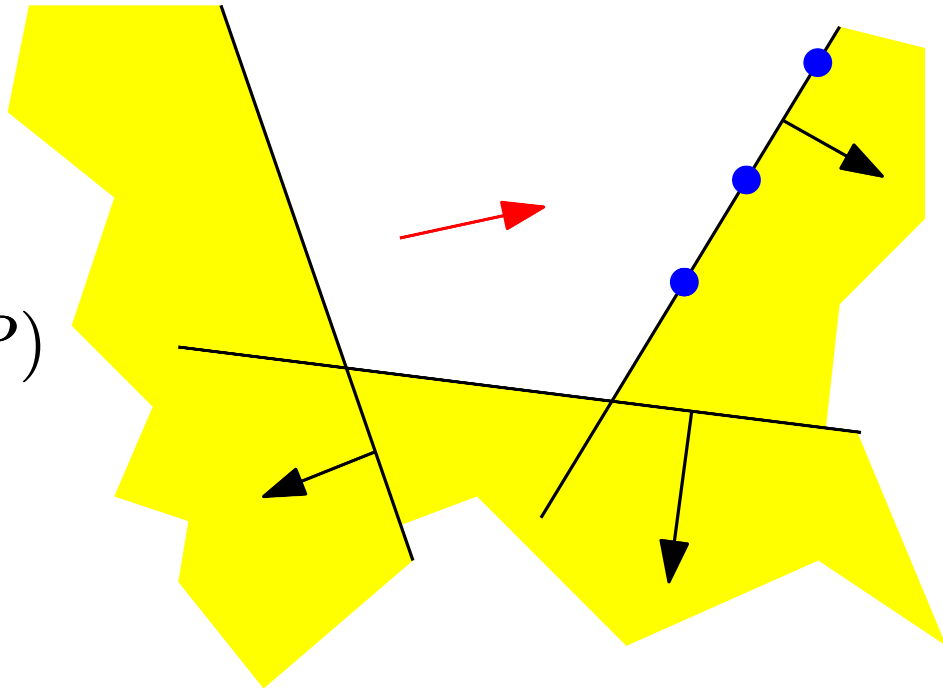
Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$ forme un polyhèdre convexe P éventuellement vide ou non borné

Le vecteur c indique la direction dans laquelle on optimise: $\max(c \cdot x \mid x \in P)$

L'optimum peut être à l'infini.

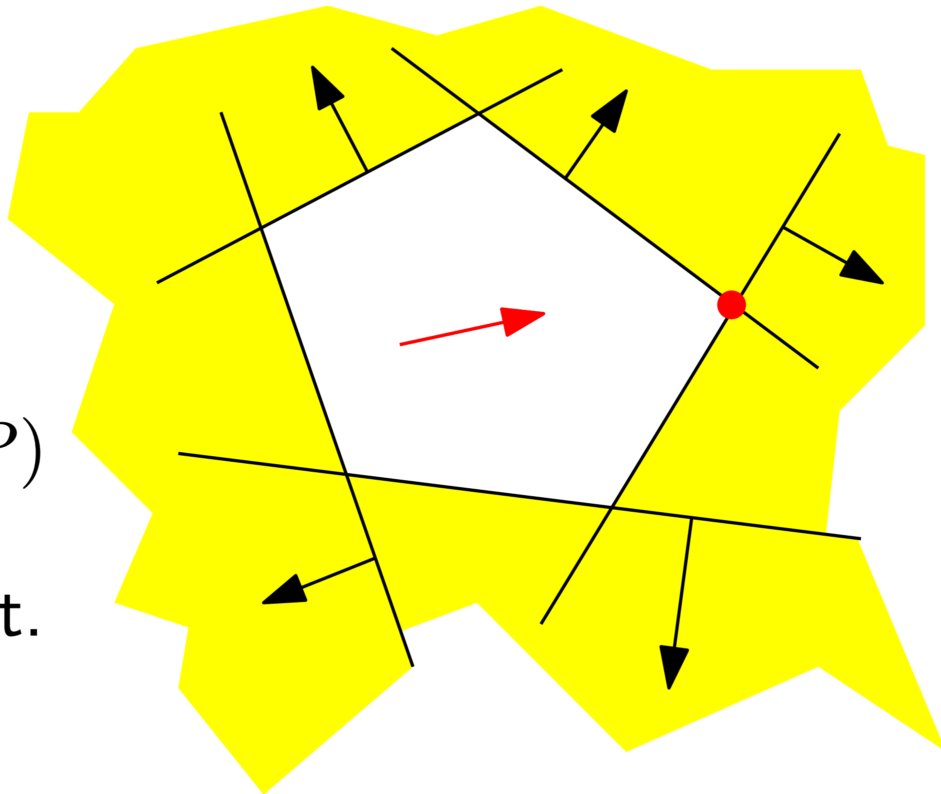


Interprétation géométrique

Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$
forme un polyhèdre convexe P
éventuellement vide ou non borné
Le vecteur c indique la direction dans
laquelle on optimise: $\max(c \cdot x \mid x \in P)$
L'optimum peut être à l'infini.
S'il est fini, il est atteint en un sommet.
C'est toujours le cas si P est borné.



Interprétation géométrique

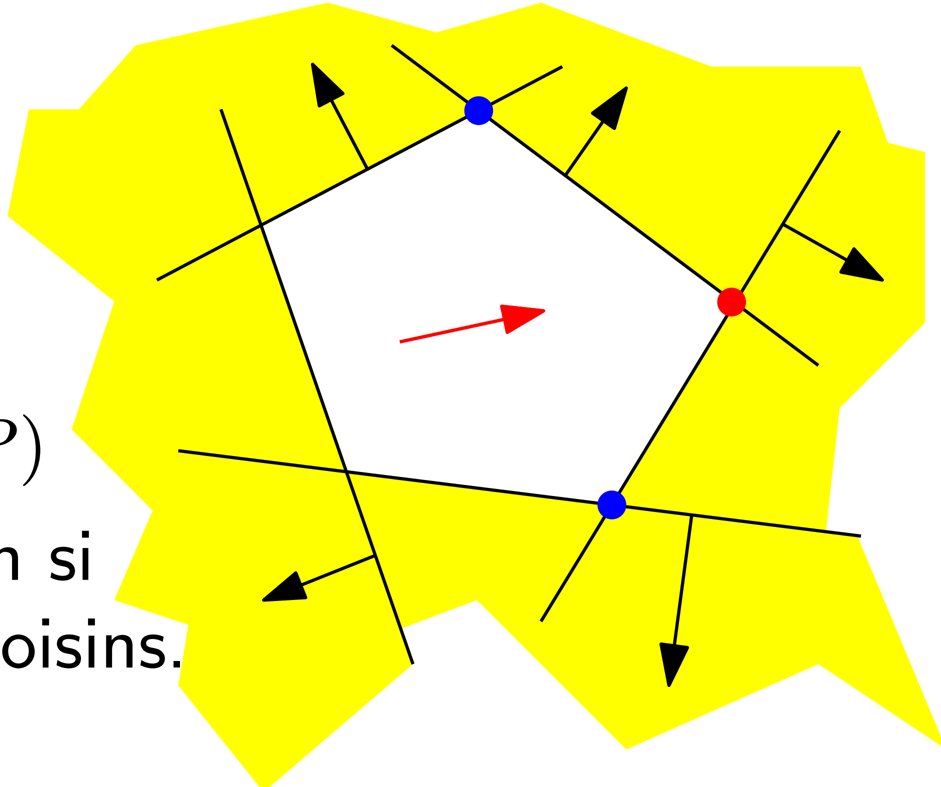
Donnée: Une matrice réelle $A = (a_{i,j})$ de taille $m \times n$,
un vecteur $b = (b_1, \ldots, b_m)$ de taille m,
un vecteur $c = (c_1, \ldots, c_n)$ de taille n.

Chaque équation $a_{i,1}x_1 + \ldots + a_{i,n}x_n \leq b_i$ coupe l'espace en 2, le long d'un l'hyperplan normal au vecteur $L_i = (a_{i,1}, \ldots, a_{i,n})$.

L'ensemble des x satisfaisant $Ax \leq b$ forme un polyhèdre convexe P éventuellement vide ou non borné

Le vecteur c indique la direction dans laquelle on optimise: $\max(c \cdot x \mid x \in P)$

Par convexité, un sommet est optimum si et seulement si il est meilleur que ses voisins.

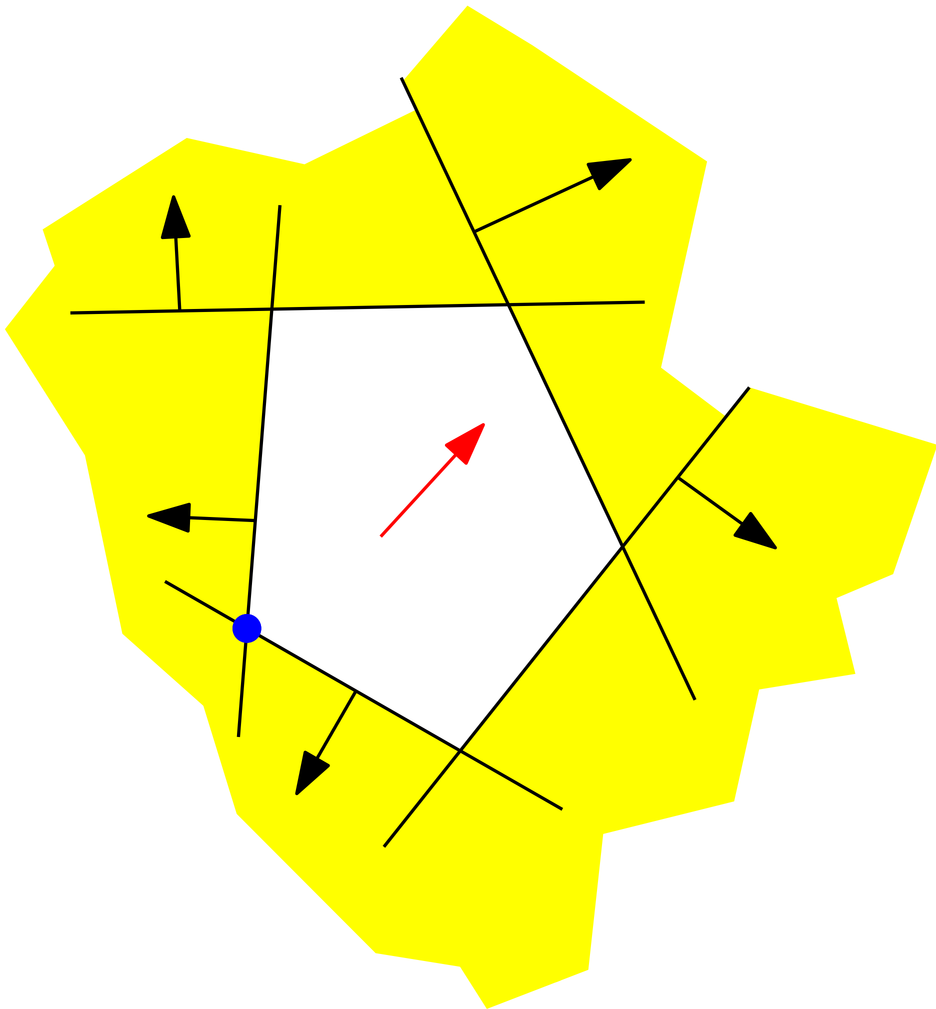


Cours 4: Programmation linéaire

- Position du problème
- Algorithme du simplexe générique
- Dualité.
- Dégénérescence et terminaison de l'algorithme

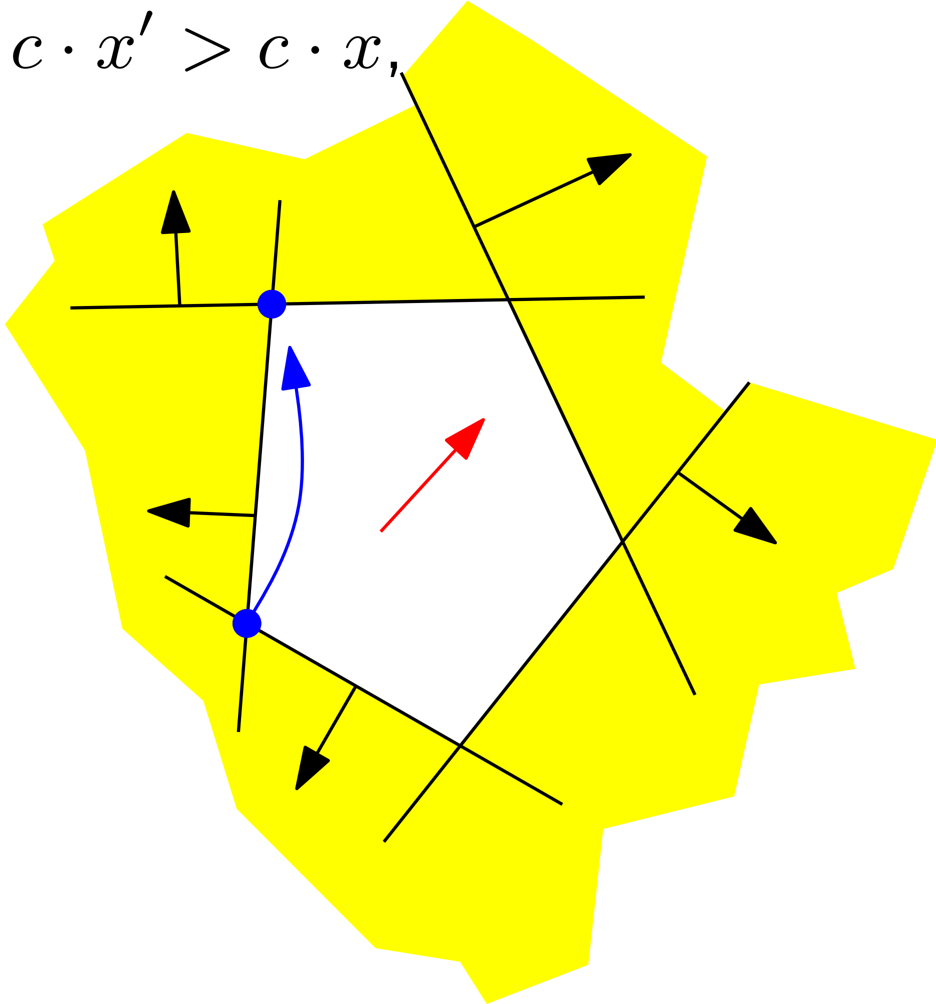
Algorithme du simplexe, premier essai

- Initialiser x avec un sommet quelconque de P



Algorithme du simplexe, premier essai

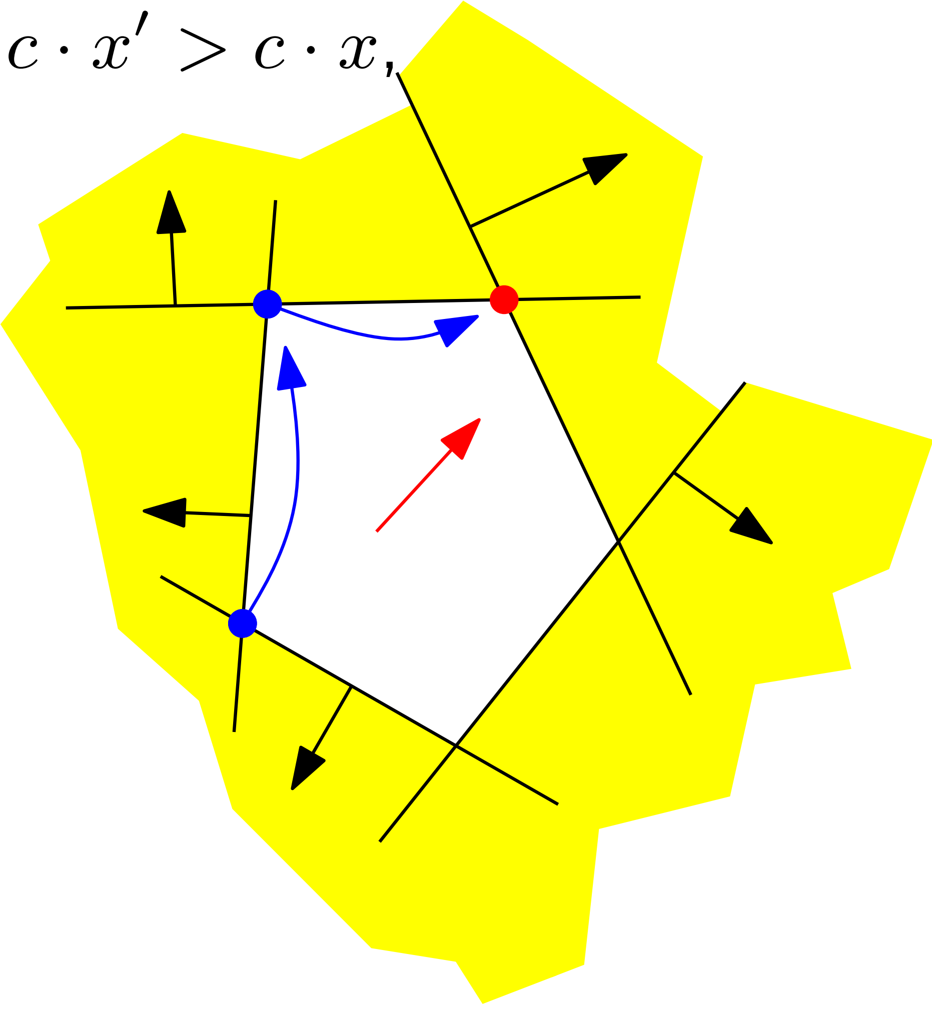
- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c{\cdot}x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.



Algorithme du simplexe, premier essai

- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c{\cdot}x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.

Remarques: Glouton...

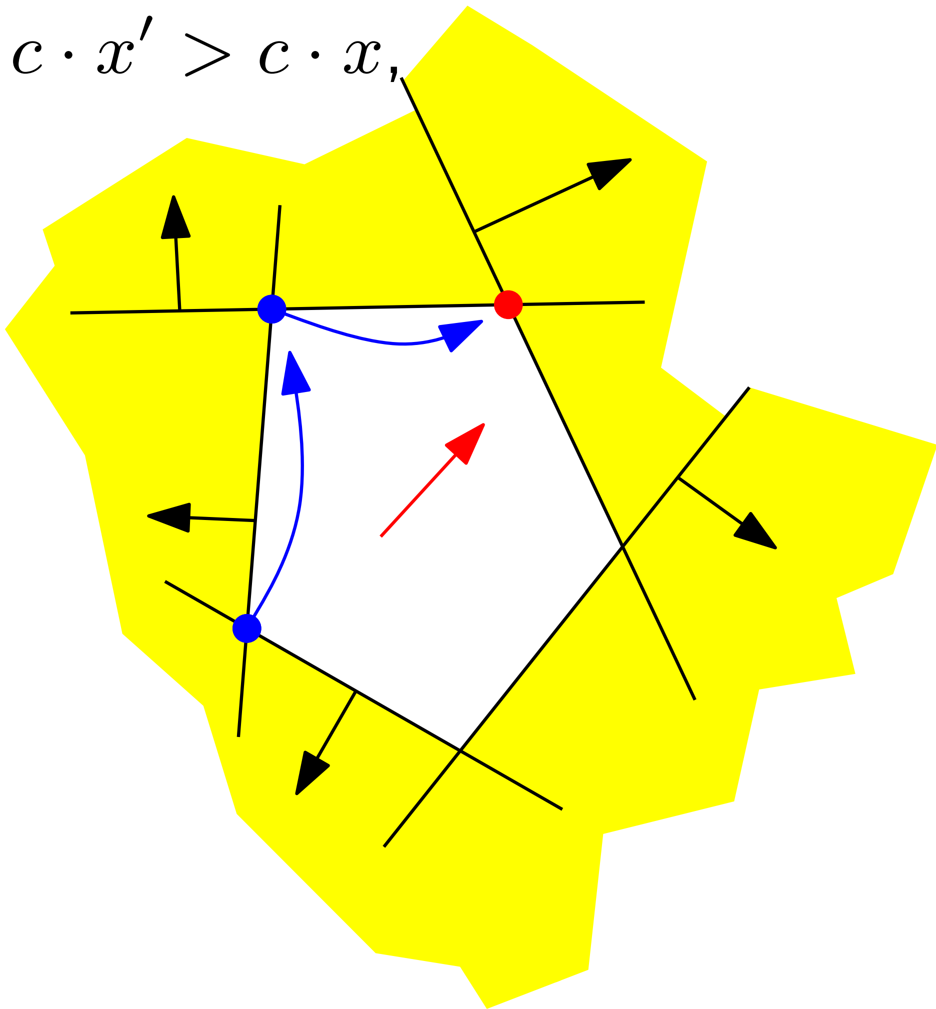


Algorithme du simplexe, premier essai

- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c{\cdot}x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.

Remarques: Glouton...

Qu'est ce qu'un sommet, un voisin ?



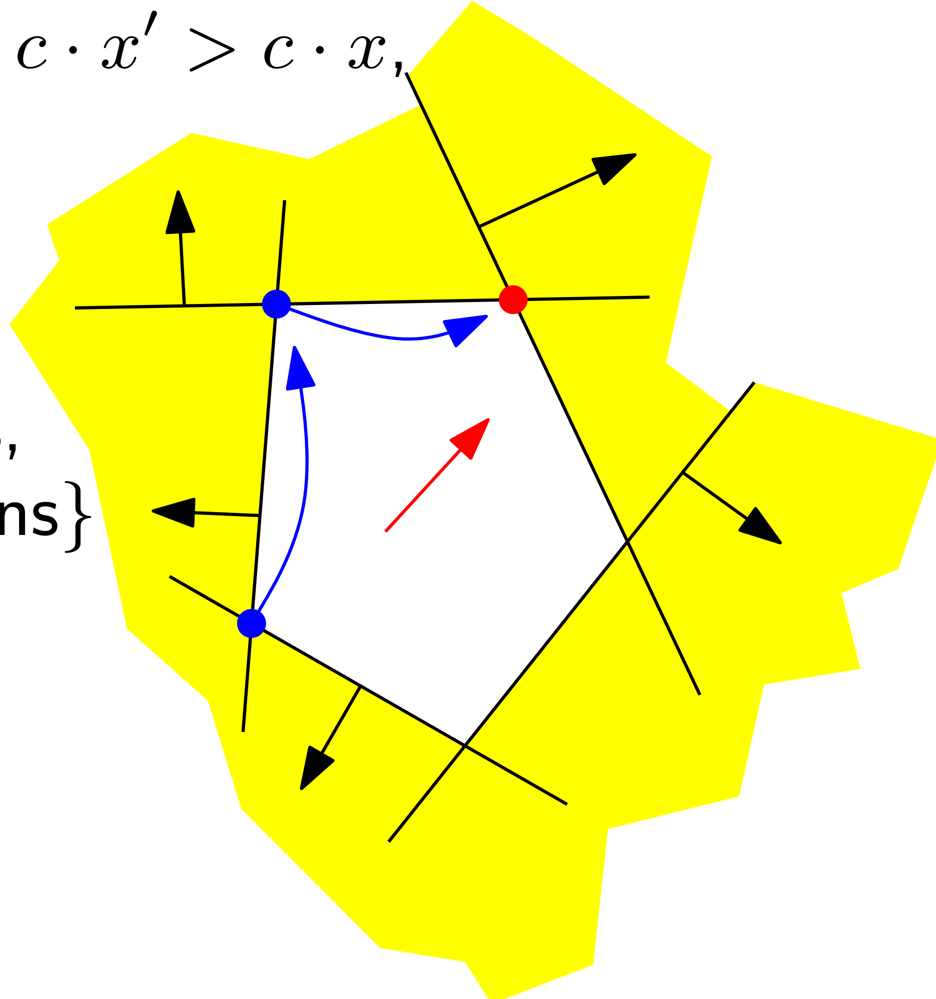
Algorithme du simplexe, premier essai

- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c \cdot x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.

Remarques: Glouton...

Qu'est ce qu'un sommet, un voisin ?

Sommet $=$ intersection de n hyperplans,
donné par $I = \{$indices des n équations$\}$



Algorithme du simplexe, premier essai

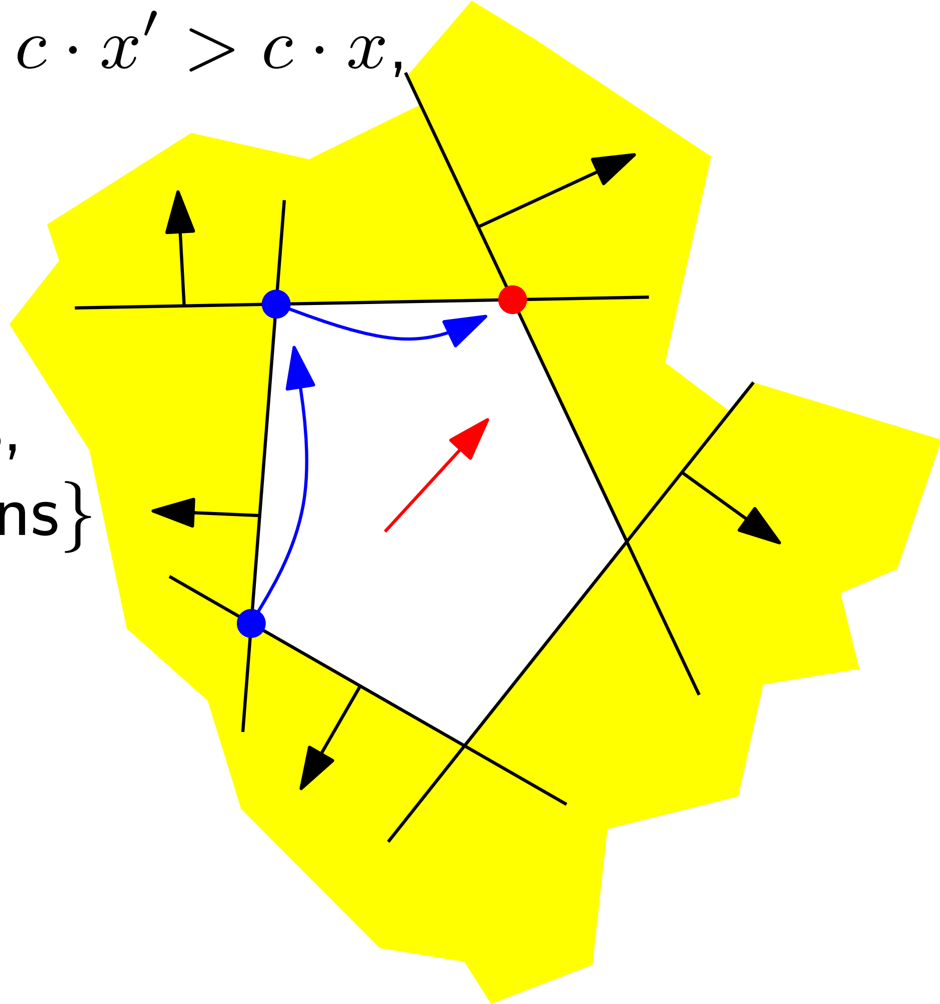
- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c \cdot x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.

Remarques: Glouton...

Qu'est ce qu'un sommet, un voisin ?

Sommet $=$ intersection de n hyperplans,
donné par $I = \{$indices des n équations$\}$

Voisin $=$ partage $n-1$ équations



Algorithme du simplexe, premier essai

- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c \cdot x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.

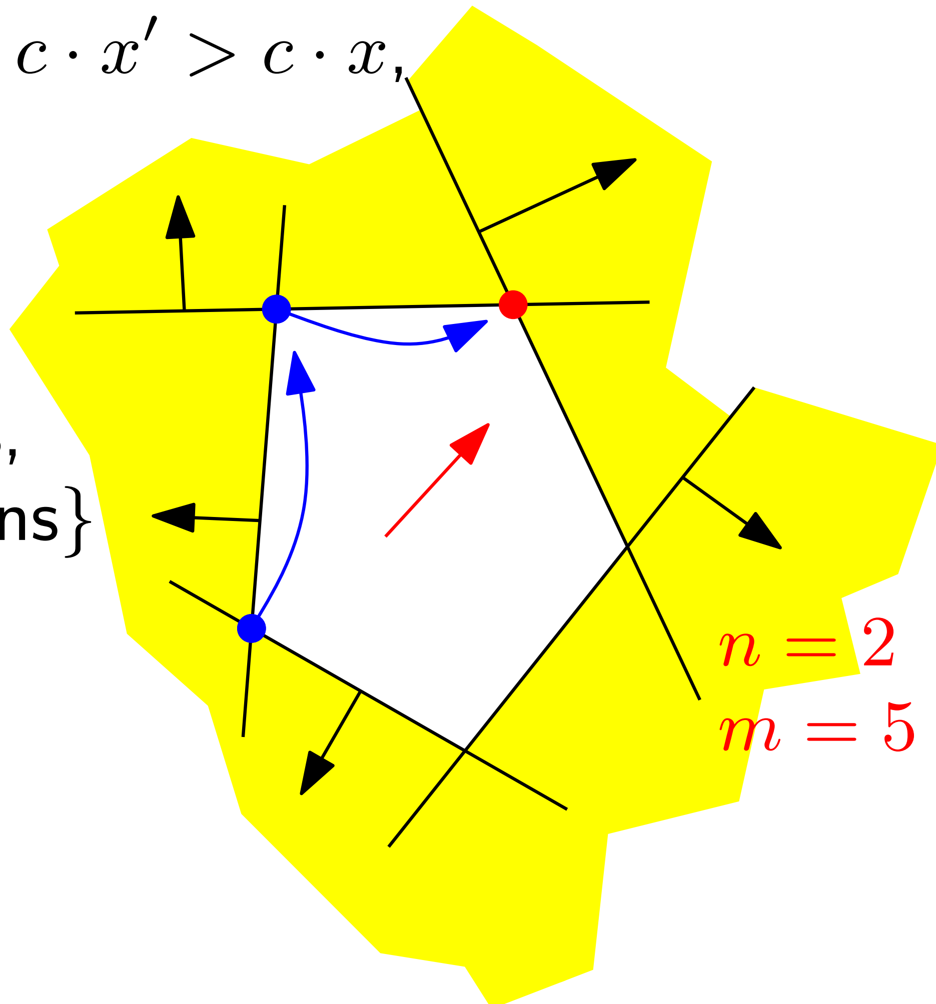
Remarques: Glouton...

Qu'est ce qu'un sommet, un voisin ?

Sommet = intersection de n hyperplans,
donné par $I = \{$indices des n équations$\}$

Voisin = partage $n-1$ équations

$\Rightarrow$ un sommet a $n(m-n)$ voisins ?



Algorithme du simplexe, premier essai

- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c{\cdot}x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.

Remarques: Glouton...

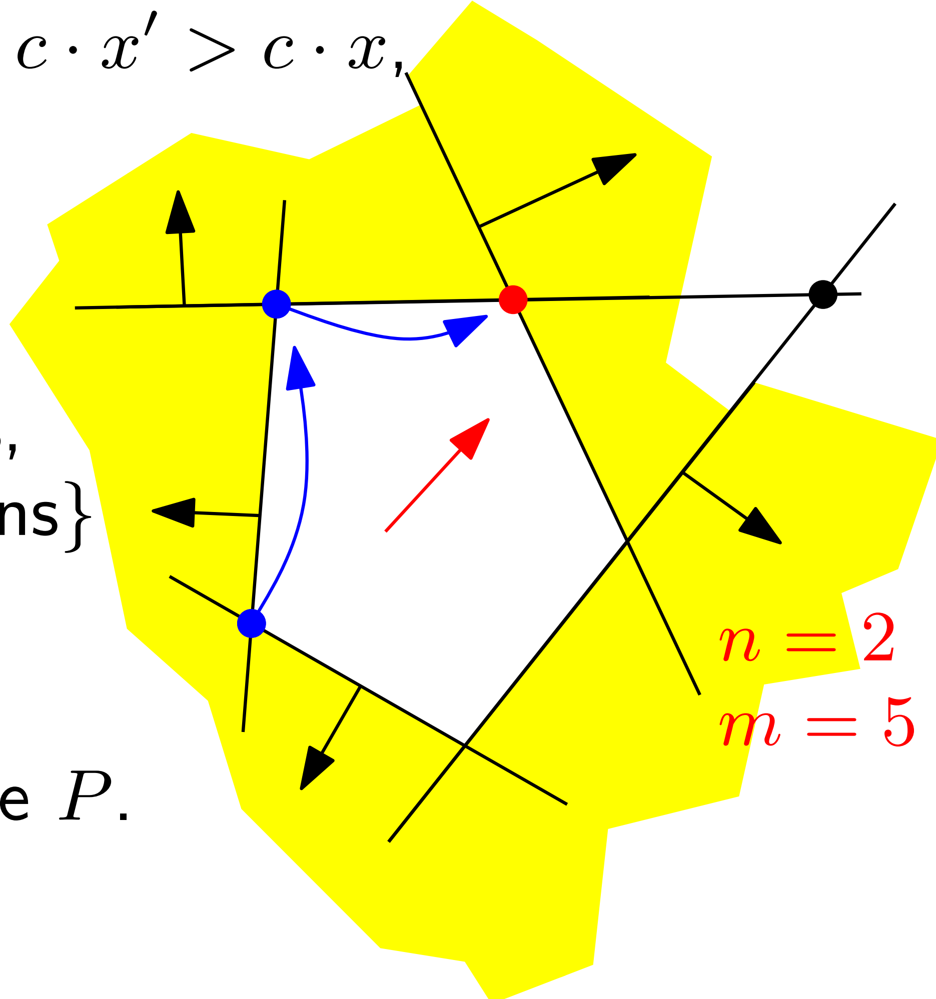
Qu'est ce qu'un sommet, un voisin ?

Sommet $=$ intersection de n hyperplans, donné par $I = \{$indices des n équations$\}$

Voisin $=$ partage $n-1$ équations

$\Rightarrow$ un sommet a $n(m-n)$ voisins ?

On veut un voisin qui soit un sommet de P.



Algorithme du simplexe, premier essai

- Initialiser x avec un sommet quelconque de P
- Tant qu'on a pas trouvé une direction infinie où $c \cdot x$ croit, et qu'il existe x' voisin de x avec $c \cdot x' > c \cdot x$, faire $x := x'$.

Remarques: Glouton...

Qu'est ce qu'un sommet, un voisin ?

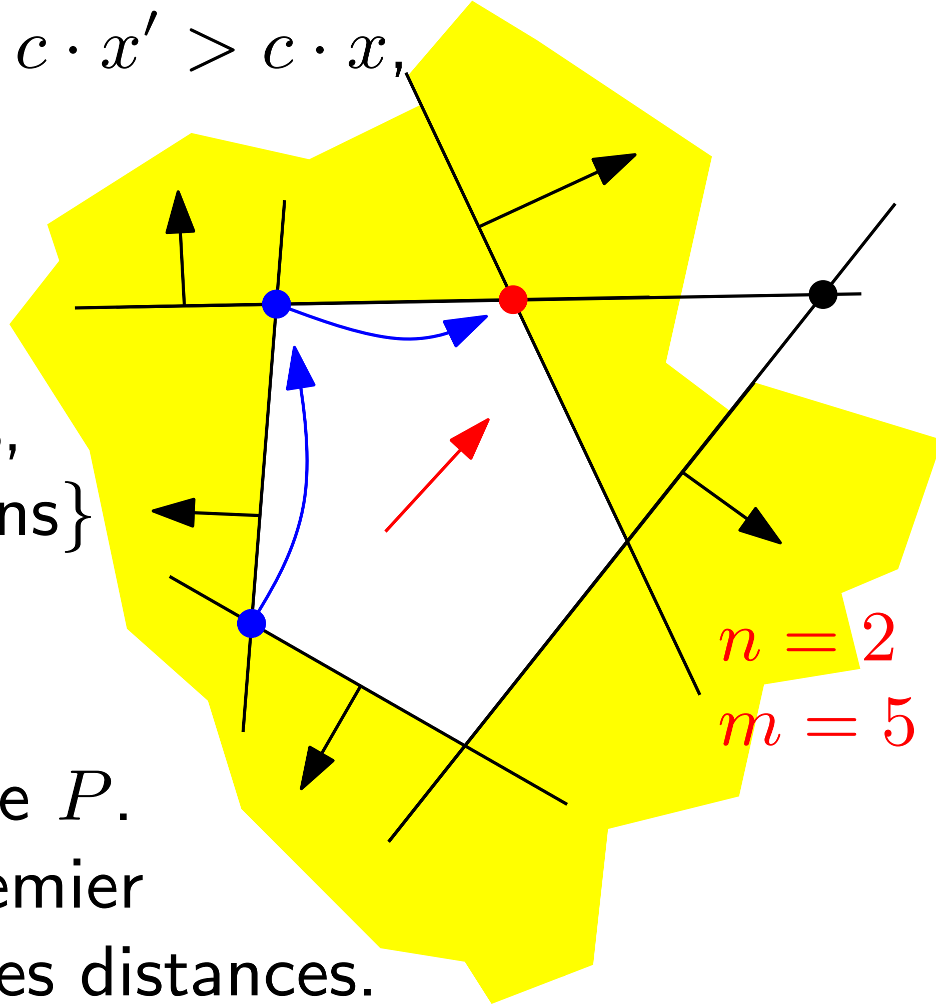
Sommet = intersection de n hyperplans,
donné par $I = \{\text{indices des } n \text{ équations}\}$

Voisin = partage $n-1$ équations

$\Rightarrow$ un sommet a $n(m-n)$ voisins ?

On veut un voisin qui soit un sommet de P.

Le long d'une arête $(I \setminus \{i\})$, seul le premier voisin rencontré est dans P: comparer les distances.

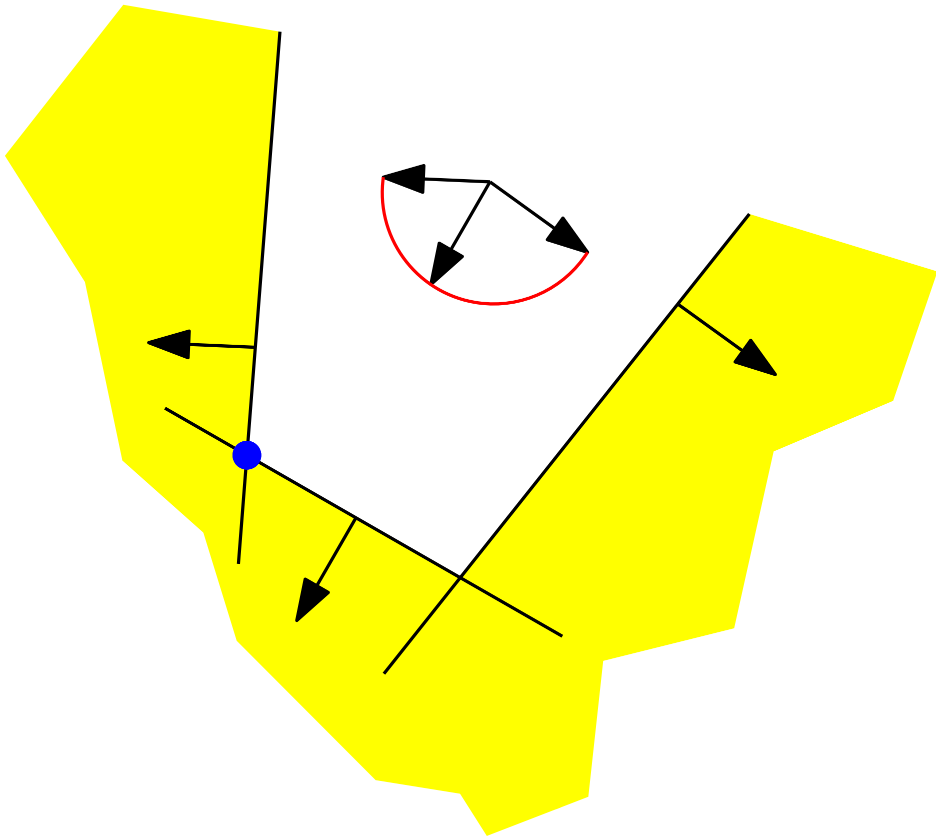


Détecter les directions infinies où $c \cdot x$ croît.

Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$
- il existe u tq $c \cdot u > 0$ et $Au \leq 0$

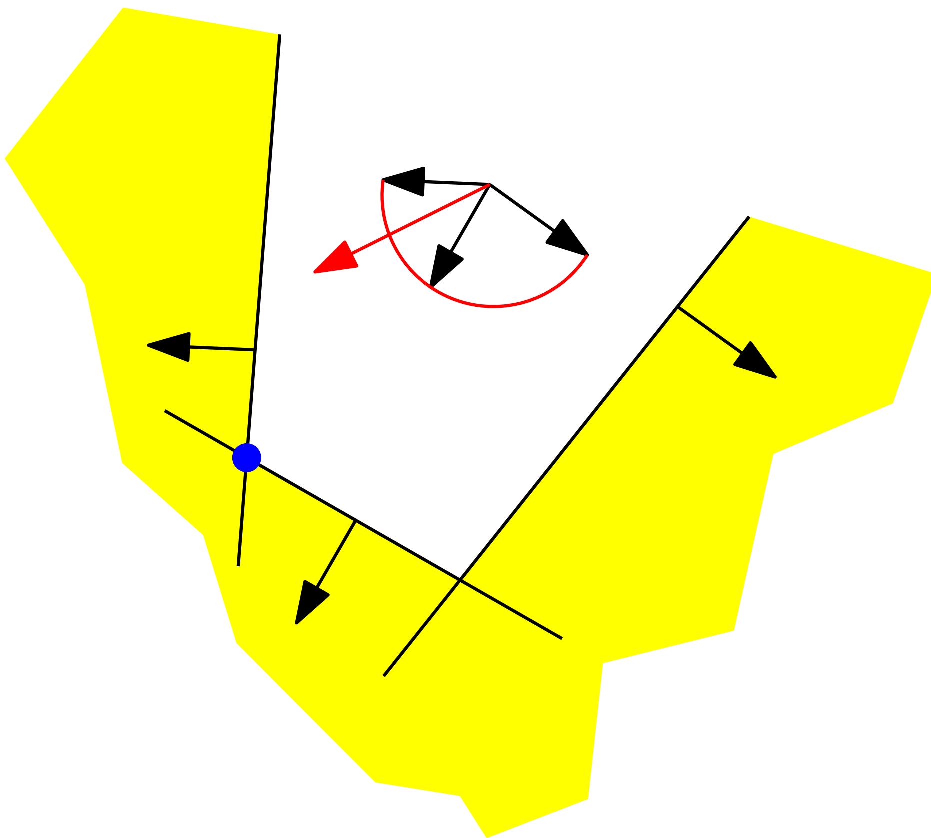


Détecter les directions infinies où $c \cdot x$ croît.

Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$
- il existe u tq $c \cdot u > 0$ et $Au \leq 0$

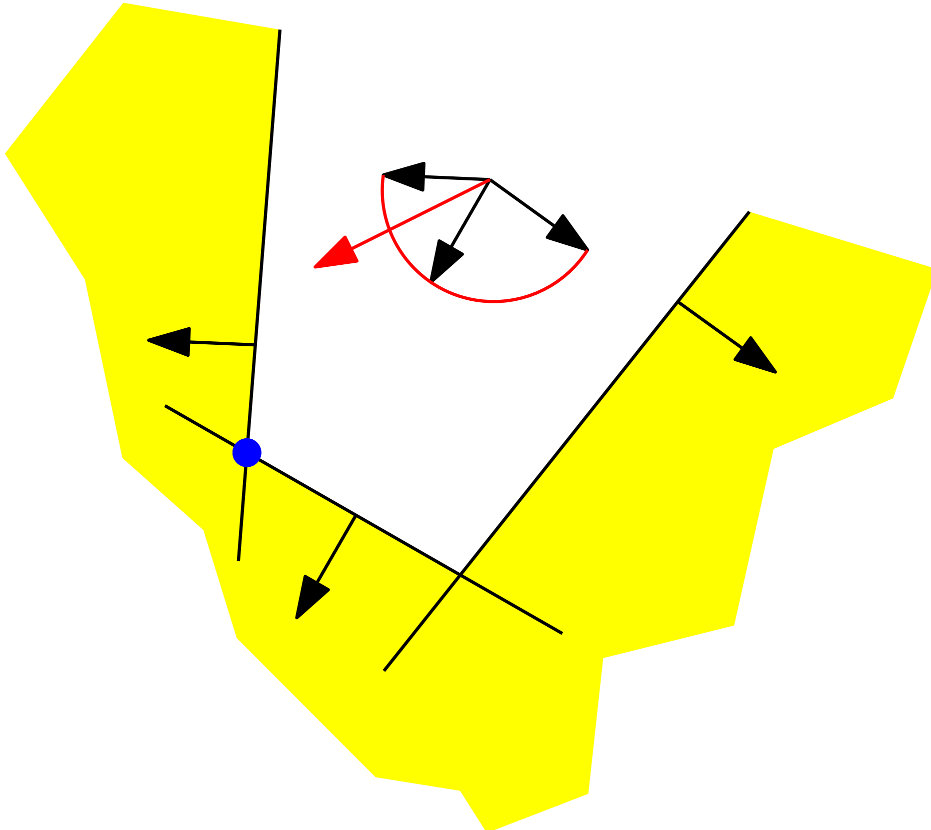


Détecter les directions infinies où $c \cdot x$ croît.

Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$ $\Rightarrow$ l'optimum est borné
- il existe u tq $c \cdot u > 0$ et $Au \leq 0$

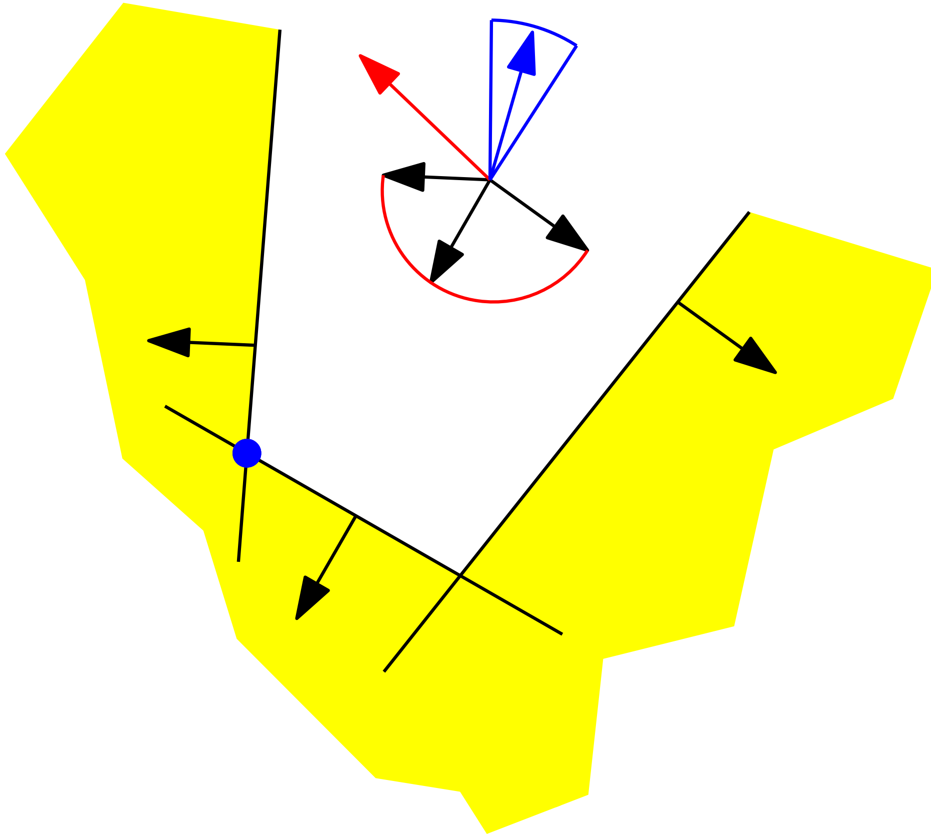


Détecter les directions infinies où $c \cdot x$ croît.

Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$ $\Rightarrow$ l'optimum est borné
- il existe u tq $c \cdot u > 0$ et $Au \leq 0$

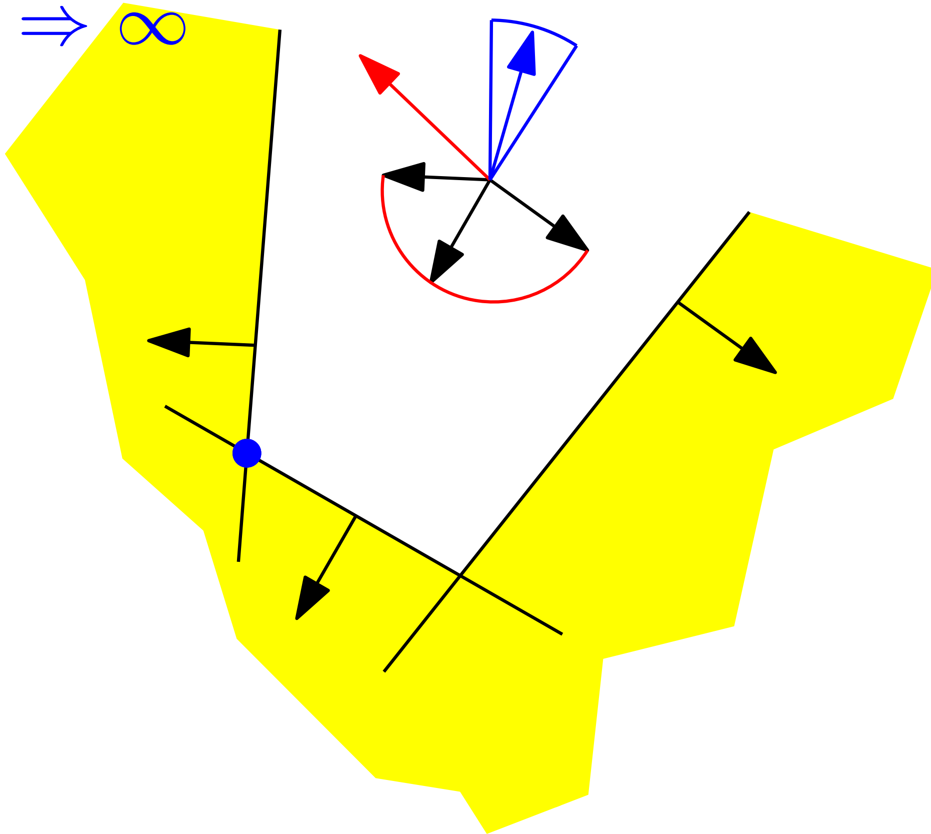


Détecter les directions infinies où $c \cdot x$ croît.

Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$ $\Rightarrow$ l'optimum est borné
- il existe u tq $c \cdot u > 0$ et $Au \leq 0$ $\Rightarrow$ ∞



Détecter les directions infinies où $c \cdot x$ croît.

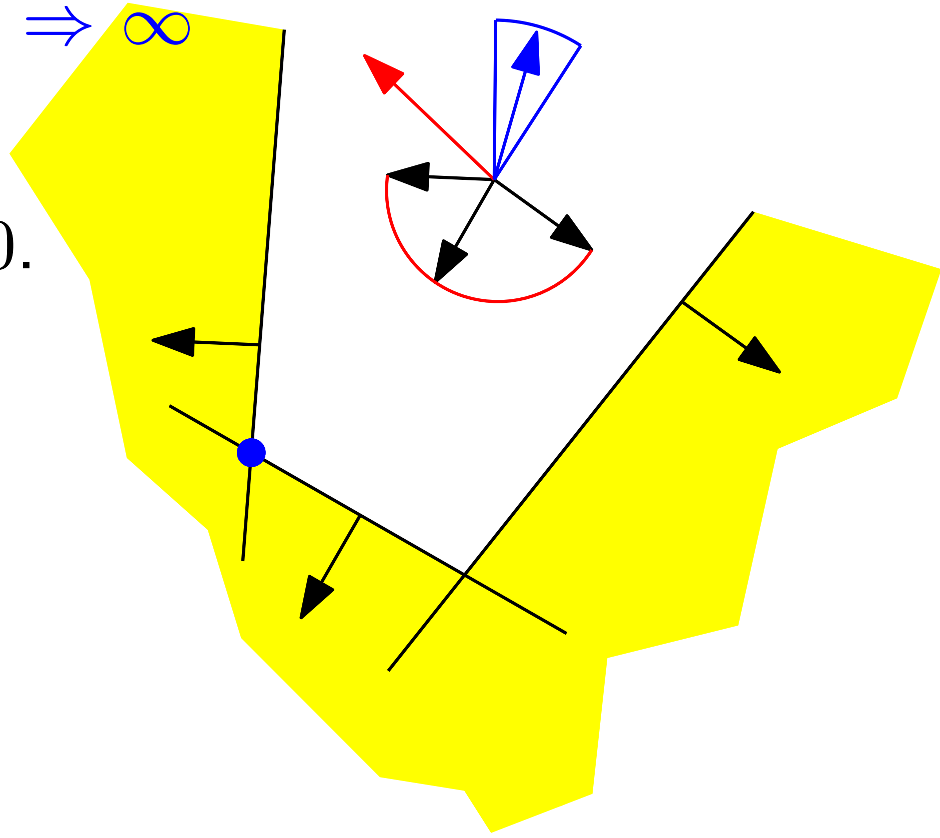
Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$ $\Rightarrow$ l'optimum est borné
- il existe u tq $c \cdot u > 0$ et $Au \leq 0$ $\Rightarrow \infty$

En effet: $c = yA \Rightarrow cu = yAu$

donc si $y \geq 0$ et $Au \leq 0$, on a $cu \leq 0$.

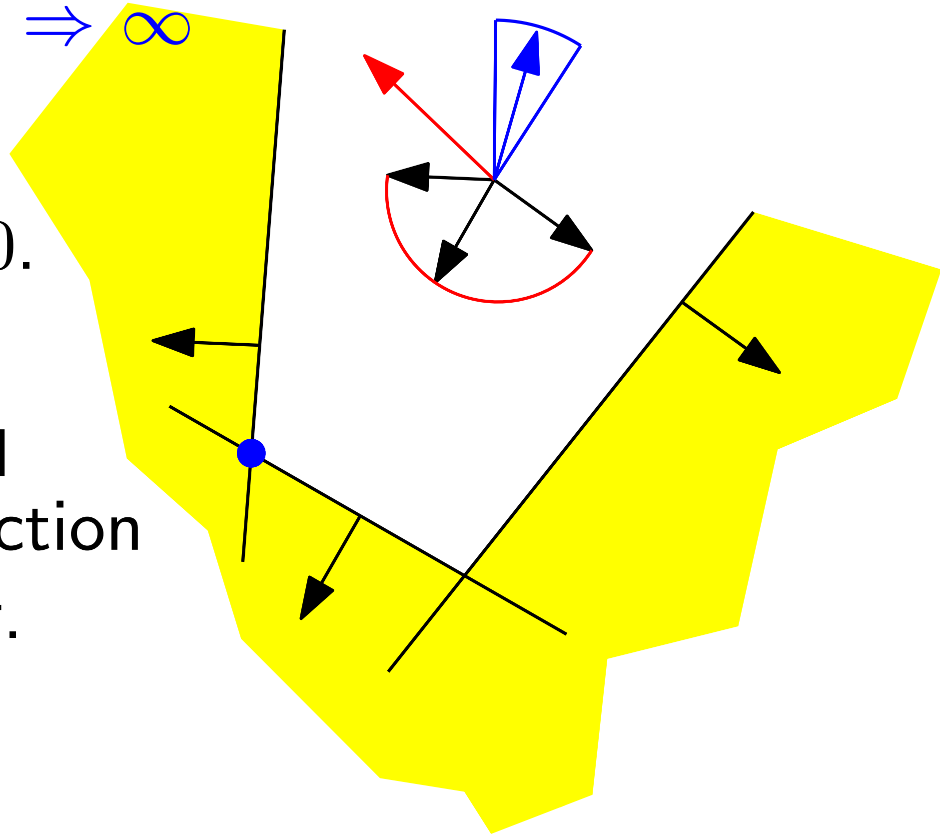


Détecter les directions infinies où $c \cdot x$ croît.

Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$ $\Rightarrow$ l'optimum est borné
- il existe u tq $c \cdot u > 0$ et $Au \leq 0 \Rightarrow \infty$



En effet: $c = yA \Rightarrow cu = yAu$
donc si $y \geq 0$ et $Au \leq 0$, on a $cu \leq 0$.

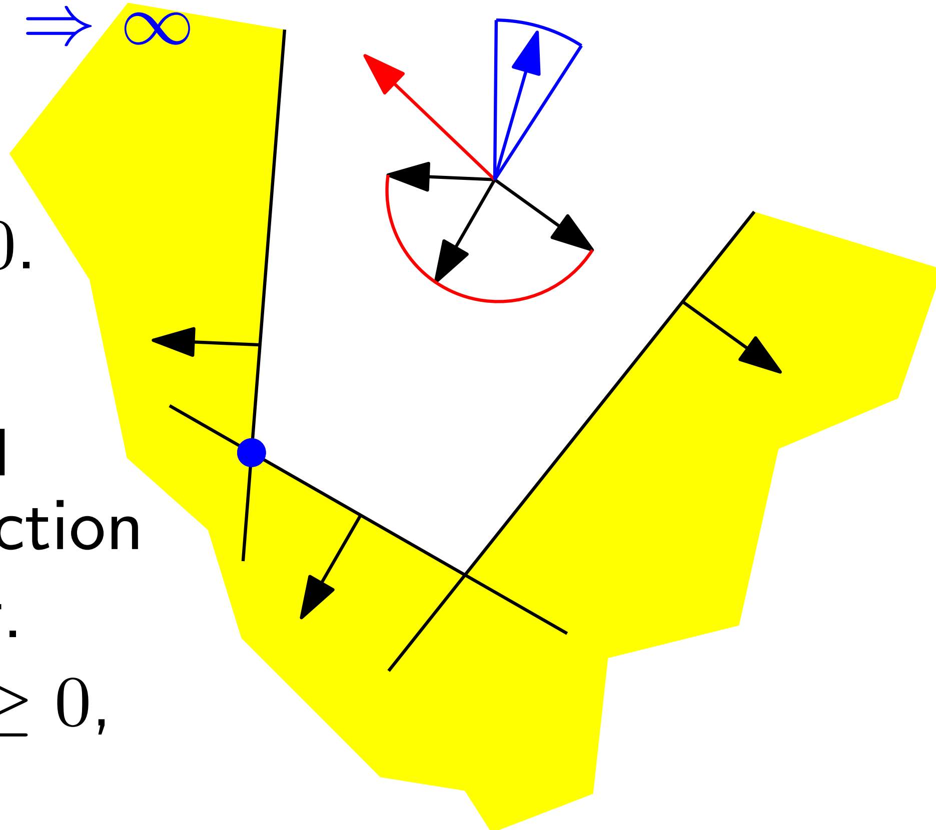
Application: si au cours de l'exécution on rencontre une arête de direction u tq $Au \leq 0$ et $cu > 0$, on a trouvé une direction de croissance infinie et on peut s'arrêter.

Détecter les directions infinies où $c \cdot x$ croît.

Si le polyhèdre n'est pas borné, $c \cdot x$ peut croître indéfiniment:

Lemme: les 2 cas suivants s'excluent mutuellement:

- c s'écrit $c = \sum_i y_i L_i$ avec les $y_i \geq 0$ $\Rightarrow$ l'optimum est borné
- il existe u tq $c \cdot u > 0$ et $Au \leq 0$ $\Rightarrow \infty$



En effet: $c = yA \Rightarrow cu = yAu$
donc si $y \geq 0$ et $Au \leq 0$, on a $cu \leq 0$.

Application: si au cours de l'exécution on rencontre une arête de direction u tq $Au \leq 0$ et $cu > 0$, on a trouvé une direction de croissance infinie et on peut s'arrêter.
Si au point x_I, $c = \sum_{i \in I} y_i L_i$ avec $y_i \geq 0$, on a trouvé l'optimum.

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k \in I}$

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k \in I}$

- Exprimer c dans la base $\{L_k\}_{k\in I}$: $c = \sum_{k \in I} y_k L_k$.

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

- Exprimer c dans la base $\{L_k\}_{k\in I}$: $c = \sum_{k\in I} y_k L_k$.

  Si $y_k \geq 0$ pour tout $k \in I$, on a trouvé l'optimum.

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k \in I}$

- Exprimer c dans la base $\{L_k\}_{k \in I}$: $c = \sum_{k \in I} y_k L_k$.
  Si $y_k \geq 0$ pour tout $k \in I$, on a trouvé l'optimum.
- Sinon on choisit le plus petit $i \in I$ tq $y_i < 0$ et on considère l'arête définie par les équations de $I' = I \setminus \{i\}$. Soit u son vecteur directeur tq $L_i u = -1$ (c'est une colonne de $(A_I)^{-1}$).

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

- Exprimer c dans la base $\{L_k\}_{k\in I}$: $c = \sum_{k\in I} y_k L_k$.

  Si $y_k \geq 0$ pour tout $k \in I$, on a trouvé l'optimum.

- Sinon on choisit le plus petit $i \in I$ tq $y_i < 0$ et on considère l'arête définie par les équations de $I' = I \setminus \{i\}$. Soit u son vecteur directeur tq $L_i u = -1$ (c'est une colonne de $(A_I)^{-1}$).

- Calculer la vitesse vers L_j quand on suit u: $v_j = L_j \cdot u$.

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

- Exprimer c dans la base $\{L_k\}_{k\in I}$: $c = \sum_{k\in I} y_k L_k$.
  Si $y_k \geq 0$ pour tout $k \in I$, on a trouvé l'optimum.
- Sinon on choisit le plus petit $i \in I$ tq $y_i < 0$ et on considère l'arête définie par les équations de $I' = I \setminus \{i\}$. Soit u son vecteur directeur tq $L_i u = -1$ (c'est une colonne de $(A_I)^{-1}$).
- Calculer la vitesse vers L_j quand on suit u: $v_j = L_j \cdot u$.
  si $v_j \leq 0$ on s'éloigne de l'hyperplan L_j: si $\forall j$, $\max = \infty$.

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

- Exprimer c dans la base $\{L_k\}_{k\in I}$: $c = \sum_{k\in I} y_k L_k$.
  Si $y_k \geq 0$ pour tout $k \in I$, on a trouvé l'optimum.
- Sinon on choisit le plus petit $i \in I$ tq $y_i < 0$ et on considère l'arête définie par les équations de $I' = I \setminus \{i\}$. Soit u son vecteur directeur tq $L_i u = -1$ (c'est une colonne de $(A_I)^{-1}$).
- Calculer la vitesse vers L_j quand on suit u: $v_j = L_j \cdot u$.
  si $v_j \leq 0$ on s'éloigne de l'hyperplan L_j: si $\forall j$, $\max = \infty$.
- Parmi les voisins sur l'arête, prendre celui qui est dans P.
  C'est celui qui est le plus proche parmi ceux du bon côté:
  donné par un des j tels que $v_j > 0$ qui minimisent $\frac{b_j - L_j x_I}{v_j}$.

Algorithme du simplexe, version *générique*

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

- Exprimer c dans la base $\{L_k\}_{k\in I}$: $c = \sum_{k\in I} y_k L_k$.
  Si $y_k \geq 0$ pour tout $k \in I$, on a trouvé l'optimum.
- Sinon on choisit le plus petit $i \in I$ tq $y_i < 0$ et on considère l'arête définie par les équations de $I' = I \setminus \{i\}$. Soit u son vecteur directeur tq $L_i u = -1$ (c'est une colonne de $(A_I)^{-1}$).
- Calculer la vitesse vers L_j quand on suit u: $v_j = L_j \cdot u$.
  si $v_j \leq 0$ on s'éloigne de l'hyperplan L_j: si $\forall j$, $\max = \infty$.
- Parmi les voisins sur l'arête, prendre celui qui est dans P.
  C'est celui qui est le plus proche parmi ceux du bon côté: donné par un des j tels que $v_j > 0$ qui minimisent $\frac{b_j - L_j x_I}{v_j}$.

  faire $I = I \setminus \{i\} \cup \{j\}$, recalculer x_I et reprendre.

Algorithme du simplexe, analyse

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

Si on avance vers le sommet x_J, $J = I \setminus \{i\} \cup \{j\}$, alors:

- $c = \sum_{k\in I} y_k L_k$, avec $y_i \geq 0$ pour $k < i$, $y_i < 0$.

Algorithme du simplexe, analyse

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

Si on avance vers le sommet x_J, $J = I \setminus \{i\} \cup \{j\}$, alors:

- $c = \sum_{k\in I} y_k L_k$, avec $y_i \geq 0$ pour $k < i$, $y_i < 0$.

Comparons l'objectif $c \cdot x_I$ avec $c \cdot x_J$:

Algorithme du simplexe, analyse

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

Si on avance vers le sommet x_J, $J = I \setminus \{i\} \cup \{j\}$, alors:

- $c = \sum_{k\in I} y_k L_k$, avec $y_i \geq 0$ pour $k < i$, $y_i < 0$.

Comparons l'objectif $c \cdot x_I$ avec $c \cdot x_J$:

- $c \cdot x_J = c \cdot x_I + c \cdot \frac{b_j - L_j x_I}{v_j} u = c \cdot x_I + \frac{b_j - L_j x_I}{v_j} c \cdot u$

  or $c \cdot u = \sum_{k\in I} y_k L_k \cdot u = -y_i > 0$ puisque u est dans les hyperplans L_k, $k \neq i$ et $L_i \cdot u = -1$.

Algorithme du simplexe, analyse

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k \in I}$

Si on avance vers le sommet x_J, $J = I \setminus \{i\} \cup \{j\}$, alors:

- $c = \sum_{k \in I} y_k L_k$, avec $y_i \geq 0$ pour $k < i$, $y_i < 0$.

Comparons l'objectif $c \cdot x_I$ avec $c \cdot x_J$:

- $c \cdot x_J = c \cdot x_I + c \cdot \frac{b_j - L_j x_I}{v_j} u = c \cdot x_I + \frac{b_j - L_j x_I}{v_j} c \cdot u$

  or $c \cdot u = \sum_{k \in I} y_k L_k \cdot u = -y_i > 0$ puisque u est dans les hyperplans L_k, $k \neq i$ et $L_i \cdot u = -1$.

$\Rightarrow$ comme prévu x_J améliore l'objectif...

Algorithme du simplexe, analyse

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k \in I}$

Si on avance vers le sommet x_J, $J = I \setminus \{i\} \cup \{j\}$, alors:

- $c = \sum_{k \in I} y_k L_k$, avec $y_i \geq 0$ pour $k < i$, $y_i < 0$.

Comparons l'objectif $c \cdot x_I$ avec $c \cdot x_J$:

- $c \cdot x_J = c \cdot x_I + c \cdot \frac{b_j - L_j x_I}{v_j} u = c \cdot x_I + \frac{b_j - L_j x_I}{v_j} c \cdot u$

  or $c \cdot u = \sum_{k \in I} y_k L_k \cdot u = -y_i > 0$ puisque u est dans les hyperplans L_k, $k \neq i$ et $L_i \cdot u = -1$.

$\Rightarrow$ comme prévu x_J améliore l'objectif... sauf si $b_j - L_j x = 0$!

Algorithme du simplexe, analyse

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$

Si on avance vers le sommet x_J, $J = I \setminus \{i\} \cup \{j\}$, alors:

- $c = \sum_{k\in I} y_k L_k$, avec $y_i \geq 0$ pour $k < i$, $y_i < 0$.

Comparons l'objectif $c \cdot x_I$ avec $c \cdot x_J$:

- $c \cdot x_J = c \cdot x_I + c \cdot \frac{b_j - L_j x_I}{v_j} u = c \cdot x_I + \frac{b_j - L_j x_I}{v_j} c \cdot u$

  or $c \cdot u = \sum_{k\in I} y_k L_k \cdot u = -y_i > 0$ puisque u est dans les hyperplans L_k, $k \neq i$ et $L_i \cdot u = -1$.

$\Rightarrow$ comme prévu x_J améliore l'objectif... sauf si $b_j - L_j x = 0$!

Si plus de n hyperplans L_k passent par x_I, certaines arêtes dégénèrent et on peut avoir $x_J = x_I$, l'algorithme risque de boucler.

Algorithme du simplexe, analyse

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k\in I}$
Si on avance vers le sommet x_J, $J = I \setminus \{i\} \cup \{j\}$, alors:

- $c = \sum_{k\in I} y_k L_k$, avec $y_i \geq 0$ pour $k < i$, $y_i < 0$.

Comparons l'objectif $c \cdot x_I$ avec $c \cdot x_J$:

- $c \cdot x_J = c \cdot x_I + c \cdot \frac{b_j - L_j x_I}{v_j} u = c \cdot x_I + \frac{b_j - L_j x_I}{v_j} c \cdot u$
  or $c \cdot u = \sum_{k\in I} y_k L_k \cdot u = -y_i > 0$ puisque u est dans les hyperplans L_k, $k \neq i$ et $L_i \cdot u = -1$.

$\Rightarrow$ comme prévu x_J améliore l'objectif... sauf si $b_j - L_j x = 0$!

Si plus de n hyperplans L_k passent par x_I, certaines arêtes dégénèrent et on peut avoir $x_J = x_I$, l'algorithme risque de boucler.

Une solution: perturber les b_i pour éliminer les dégénérescences.

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\text{argmax}(cx \mid Ax \leq b)$

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\mathrm{argmax}(cx \mid Ax \leq b)$

On peut toujours poser $x_i = y_i - z_i$ (le nombre de variables double) et maximiser $(c, -c) \cdot \binom{y}{z}$ pour $(A, -A)\binom{y}{z} \leq b$ et $\binom{y}{z} \geq 0$.

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\mathrm{argmax}(cx \mid Ax \leq b)$

On peut toujours poser $x_i = y_i - z_i$ (le nombre de variables double) et maximiser $(c, -c) \cdot \binom{y}{z}$ pour $(A, -A)\binom{y}{z} \leq b$ et $\binom{y}{z} \geq 0$.

On cherche donc: $\mathrm{argmax}(cx \mid Ax \leq b,\ x \geq 0)$

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\mathrm{argmax}(cx \mid Ax \leq b)$

On peut toujours poser $x_i = y_i - z_i$ (le nombre de variables double) et maximiser $(c, -c) \cdot \binom{y}{z}$ pour $(A, -A)\binom{y}{z} \leq b$ et $\binom{y}{z} \geq 0$.

On cherche donc: $\mathrm{argmax}(cx \mid Ax \leq b,\ x \geq 0)$

Si les b_i sont positifs, $(0, \ldots, 0)$ est notre sommet initial.

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\mathrm{argmax}(cx \mid Ax \leq b)$

On peut toujours poser $x_i = y_i - z_i$ (le nombre de variables double) et maximiser $(c, -c) \cdot \binom{y}{z}$ pour $(A, -A)\binom{y}{z} \leq b$ et $\binom{y}{z} \geq 0$.

On cherche donc: $\mathrm{argmax}(cx \mid Ax \leq b,\ x \geq 0)$

Si les b_i sont positifs, $(0, \ldots, 0)$ est notre sommet initial.

Sinon on cherche un sommet de $P = \{x \mid Ax \leq b,\ x \geq 0\}$

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\mathrm{argmax}(cx \mid Ax \leq b)$

On peut toujours poser $x_i = y_i - z_i$ (le nombre de variables double) et maximiser $(c, -c) \cdot \binom{y}{z}$ pour $(A, -A)\binom{y}{z} \leq b$ et $\binom{y}{z} \geq 0$.

On cherche donc: $\mathrm{argmax}(cx \mid Ax \leq b,\ x \geq 0)$

Si les b_i sont positifs, $(0, \ldots, 0)$ est notre sommet initial.

Sinon on cherche un sommet de $P = \{x \mid Ax \leq b,\ x \geq 0\}$

On fait cela en cherchant parmi les $(x_1, \ldots, x_n, t)$ tq

$$a_{i,1}x_1 + \ldots + a_{i,n}x_n - t \leq b_i \text{ pour tout } i,$$

un point qui minimise t.

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\text{argmax}(cx \mid Ax \leq b)$

On peut toujours poser $x_i = y_i - z_i$ (le nombre de variables double) et maximiser $(c, -c) \cdot \binom{y}{z}$ pour $(A, -A)\binom{y}{z} \leq b$ et $\binom{y}{z} \geq 0$.

On cherche donc: $\text{argmax}(cx \mid Ax \leq b,\ x \geq 0)$

Si les b_i sont positifs, $(0, \ldots, 0)$ est notre sommet initial.

Sinon on cherche un sommet de $P = \{x \mid Ax \leq b,\ x \geq 0\}$

On fait cela en cherchant parmi les $(x_1, \ldots, x_n, t)$ tq

$$a_{i,1}x_1 + \ldots + a_{i,n}x_n - t \leq b_i \text{ pour tout } i,$$

un point qui minimise t.

On connait un sommet initial $(0, \ldots, 0, T)$ pour ce problème linéaire.

Algorithme du simplexe, initialisation

Pour démarrer l'algorithme il faut avoir un sommet de P.

On cherche: $\mathrm{argmax}(cx \mid Ax \leq b)$

On peut toujours poser $x_i = y_i - z_i$ (le nombre de variables double) et maximiser $(c, -c) \cdot \binom{y}{z}$ pour $(A, -A)\binom{y}{z} \leq b$ et $\binom{y}{z} \geq 0$.

On cherche donc: $\mathrm{argmax}(cx \mid Ax \leq b,\ x \geq 0)$

Si les b_i sont positifs, $(0, \ldots, 0)$ est notre sommet initial.

Sinon on cherche un sommet de $P = \{x \mid Ax \leq b,\ x \geq 0\}$

On fait cela en cherchant parmi les $(x_1, \ldots, x_n, t)$ tq

$$a_{i,1}x_1 + \ldots + a_{i,n}x_n - t \leq b_i \text{ pour tout } i,$$

un point qui minimise t.

On connait un sommet initial $(0, \ldots, 0, T)$ pour ce problème linéaire.

Il existe (et on obtient) un sommet de P ssi $t = 0$ dans sa solution.

Cours 4: Programmation linéaire

- Position du problème
- Algorithme du simplexe générique
- Dualité.
- Dégénérescence et terminaison de l'algorithme

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \leq b,\ x \geq 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \geq c,\ y \geq 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \geq c,\ y \geq 0) \ = \ \max(cx \mid Ax \leq b, x \geq 0)$$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \leq b,\ x \geq 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \geq c,\ y \geq 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \geq c,\ y \geq 0) \ = \ \max(cx \mid Ax \leq b, x \geq 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{gathered}\max(4x + 5x') \\ 10x + 10x' \leq 50 \\ 10x + 20x' \leq 80 \\ 20x + 10x' \leq 80 \\ x \geq 0,\ x' \geq 0\end{gathered}$$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \leq b,\ x \geq 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \geq c,\ y \geq 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \geq c,\ y \geq 0) \;=\; \max(cx \mid Ax \leq b, x \geq 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{array}{ll} \times & \max(4x + 5x') \\ y_1 & 10x + 10x' \leq 50 \\ y_2 & 10x + 20x' \leq 80 \\ y_3 & 20x + 10x' \leq 80 \\ & x \geq 0,\ x' \geq 0 \end{array}$$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \leq b,\ x \geq 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \geq c,\ y \geq 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \geq c,\ y \geq 0) \ = \ \max(cx \mid Ax \leq b, x \geq 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{array}{ll} \times & \max(4x+5x') \\ y_1 & 10x+10x' \leq 50 \\ y_2 & 10x+20x' \leq 80 \\ y_3 & 20x+10x' \leq 80 \\ & x \geq 0,\ x' \geq 0 \end{array}$$

$$y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0.$$

$$\Rightarrow \qquad (10y_1+10y_2+20y_3)x + (10y_1+20y_2+10y_3)x' \leq 50y_1+80y_2+80y_3$$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \le b,\ x \ge 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \ge c,\ y \ge 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \ge c,\ y \ge 0) \ = \ \max(cx \mid Ax \le b, x \ge 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{array}{ll} \times & \max(4x + 5x') \\ y_1 & 10x + 10x' \le 50 \\ y_2 & 10x + 20x' \le 80 \\ y_3 & 20x + 10x' \le 80 \\ & x \ge 0,\ x' \ge 0 \end{array}$$

$$\begin{array}{l} 10y_1 + 10y_2 + 20y_3 \ge 4 \\ 10y_1 + 20y_2 + 10y_3 \ge 5 \\ y_1 \ge 0,\ y_2 \ge 0,\ y_3 \ge 0. \end{array}$$

$\Rightarrow 4x + 5x' \le (10y_1 + 10y_2 + 20y_3)x + (10y_1 + 20y_2 + 10y_3)x' \le 50y_1 + 80y_2 + 80y_3$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \leq b,\ x \geq 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \geq c,\ y \geq 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \geq c,\ y \geq 0) \ = \ \max(cx \mid Ax \leq b, x \geq 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{array}{ll} \times & \max(4x + 5x') \\ y_1 & 10x + 10x' \leq 50 \\ y_2 & 10x + 20x' \leq 80 \\ y_3 & 20x + 10x' \leq 80 \\ & x \geq 0,\ x' \geq 0 \end{array}$$

$$\begin{array}{c} \min(50y_1 + 80y_2 + 80y_3) \\ 10y_1 + 10y_2 + 20y_3 \geq 4 \\ 10y_1 + 20y_2 + 10y_3 \geq 5 \\ y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0. \end{array}$$

$\Rightarrow 4x + 5x' \leq (10y_1 + 10y_2 + 20y_3)x + (10y_1 + 20y_2 + 10y_3)x' \leq 50y_1 + 80y_2 + 80y_3$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \leq b,\ x \geq 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \geq c,\ y \geq 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \geq c,\ y \geq 0) \ = \ \max(cx \mid Ax \leq b, x \geq 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{array}{ll} \times & \max(4x + 5x') \\ y_1 & 10x + 10x' \leq 50 \\ y_2 & 10x + 20x' \leq 80 \\ y_3 & 20x + 10x' \leq 80 \\ & x \geq 0,\ x' \geq 0 \end{array}$$

Dual

$$\begin{array}{c} \min(50y_1 + 80y_2 + 80y_3) \\ 10y_1 + 10y_2 + 20y_3 \geq 4 \\ 10y_1 + 20y_2 + 10y_3 \geq 5 \\ y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0. \end{array}$$

$\Rightarrow$ $4x + 5x' \leq (10y_1 + 10y_2 + 20y_3)x + (10y_1 + 20y_2 + 10y_3)x' \leq 50y_1 + 80y_2 + 80y_3$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \le b,\ x \ge 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \ge c,\ y \ge 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \ge c,\ y \ge 0) \ = \ \max(cx \mid Ax \le b, x \ge 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{array}{ll} \times & \max(4x+5x') \\ y_1 & 10x+10x' \le 50 \\ y_2 & 10x+20x' \le 80 \\ y_3 & 20x+10x' \le 80 \\ & x \ge 0,\ x' \ge 0 \end{array}$$

Le théorème dit qu'on peut obtenir une borne optimale par multiplicateur.

Dual

$$\begin{array}{c} \min(50y_1+80y_2+80y_3) \\ 10y_1+10y_2+20y_3 \ge 4 \\ 10y_1+20y_2+10y_3 \ge 5 \\ y_1 \ge 0,\ y_2 \ge 0,\ y_3 \ge 0. \end{array}$$

$$\Rightarrow 4x+5x' \le (10y_1+10y_2+20y_3)x + (10y_1+20y_2+10y_3)x' \le 50y_1+80y_2+80y_3$$

Primal/dual et théorème de dualité

Théorème (cas inégalités+positivité). Si l'un des 2 problèmes suivant

$$(\mathcal{P}) \qquad \max(cx \mid Ax \leq b,\ x \geq 0)$$

$$(\mathcal{D}) \qquad \min(yb \mid yA \geq c,\ y \geq 0)$$

admet une solution finie, alors l'autre aussi et on a

$$\min(yb \mid yA \geq c,\ y \geq 0) \;=\; \max(cx \mid Ax \leq b, x \geq 0)$$

Interprétation. Retour au fleuriste.

Primal

$$\begin{array}{ll} \times & \max(4x + 5x') \\ y_1 & 10x + 10x' \leq 50 \\ y_2 & 10x + 20x' \leq 80 \\ y_3 & 20x + 10x' \leq 80 \\ & x \geq 0,\ x' \geq 0 \end{array}$$

Le théorème dit qu'on peut obtenir une borne optimale par multiplicateur.

Si on augmente la denrée i de Δ_i, le gain augmente de $\Delta_i y_i$.

Dual

$$\begin{array}{l} \min(50y_1 + 80y_2 + 80y_3) \\ 10y_1 + 10y_2 + 20y_3 \geq 4 \\ 10y_1 + 20y_2 + 10y_3 \geq 5 \\ y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0. \end{array}$$

$\Rightarrow 4x + 5x' \leq (10y_1 + 10y_2 + 20y_3)x + (10y_1 + 20y_2 + 10y_3)x' \leq 50y_1 + 80y_2 + 80y_3$

Primal/dual et théorème de dualité

Primal / Dual, cas général. Le théorème s'applique aux paires:

Primal	Dual
$\max(c_1x_1 + \ldots + c_nx_n)$	$\min(b_1y_1 + \ldots + b_my_m)$
$a_{i1}x_1 + \ldots a_{in}x_n \leq b_i$ pour $i \in I$	$a_{1j}y_1 + \ldots a_{mj}y_m \geq c_j$ pour $j \in P$
$a_{i1}x_1 + \ldots a_{in}x_n = b_i$ pour $i \in E$	$a_{1j}y_1 + \ldots a_{mj}y_m = c_j$ pour $j \in N$
$x_j \geq 0$ pour $j \in P$.	$y_i \geq 0$ pour $i \in I$.

où $m = |I| + |E|$ et $n = |P| + |N|$

Primal/dual et théorème de dualité

Primal / Dual, cas général. Le théorème s'applique aux paires:

Primal	Dual
$\max(c_1x_1 + \ldots + c_nx_n)$	$\min(b_1y_1 + \ldots + b_my_m)$
$a_{i1}x_1 + \ldots a_{in}x_n \leq b_i$ pour $i \in I$	$a_{1j}y_1 + \ldots a_{mj}y_m \geq c_j$ pour $j \in P$
$a_{i1}x_1 + \ldots a_{in}x_n = b_i$ pour $i \in E$	$a_{1j}y_1 + \ldots a_{mj}y_m = c_j$ pour $j \in N$
$x_j \geq 0$ pour $j \in P$.	$y_i \geq 0$ pour $i \in I$.

où $m = |I| + |E|$ et $n = |P| + |N|$

Les inégalité donnent dans le dual des variables contraintes à être positives, les égalités des variables quelconques (penser à l'interprétation des variables duales comme multiplicateurs).

Primal/dual et théorème de dualité

Primal / Dual, cas général. Le théorème s'applique aux paires:

Primal	Dual
$\max(c_1x_1+\ldots+c_nx_n)$	$\min(b_1y_1+\ldots+b_my_m)$
$a_{i1}x_1+\ldots a_{in}x_n\leq b_i$ pour $i\in I$	$a_{1j}y_1+\ldots a_{mj}y_m\geq c_j$ pour $j\in P$
$a_{i1}x_1+\ldots a_{in}x_n=b_i$ pour $i\in E$	$a_{1j}y_1+\ldots a_{mj}y_m=c_j$ pour $j\in N$
$x_j\geq 0$ pour $j\in P$.	$y_i\geq 0$ pour $i\in I$.

où $m=\|I\|+\|E\|$ et $n=\|P\|+\|N\|$

Les inégalité donnent dans le dual des variables contraintes à être positives, les égalités des variables quelconques (penser à l'interprétation des variables duales comme multiplicateurs).

En TD: le théorème de dualité $\Rightarrow$ $\mathrm{maxflow}=\mathrm{mincut}$

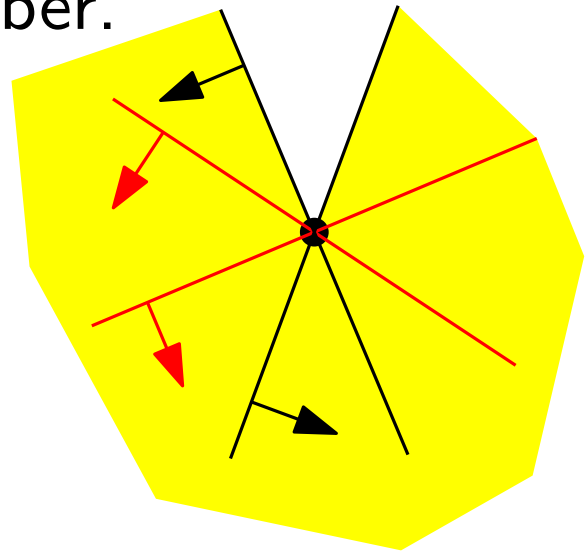
Cours 4: Programmation linéaire

- Position du problème
- Algorithme du simplexe générique
- Dualité.
- Dégénérescence et terminaison de l'algorithme

Dégénérescence et risque de bouclage

Dégénérescence: un sommet appartient à plus de n hyperplans.

Il n'est pas forcément acceptable de perturber.



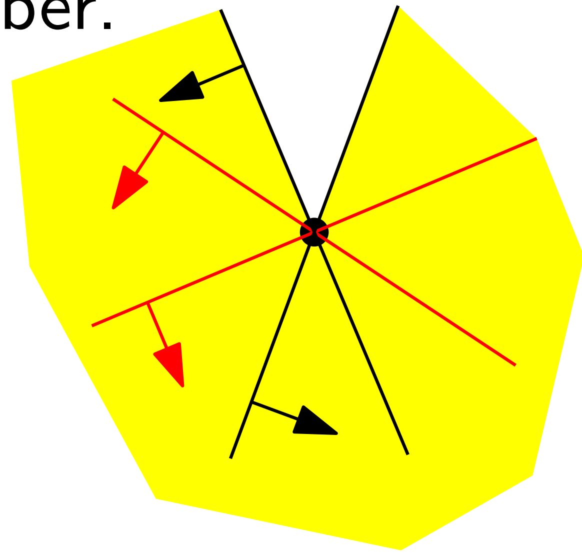
Dégénérescence et risque de bouclage

Dégénérescence: un sommet appartient à plus de n hyperplans.

Il n'est pas forcément acceptable de perturber.

Parmi les voisins de $x_{\{2,3\}}$ il y a

$x_{\{1,3\}}$, $x_{\{1,4\}}$, $x_{\{1,2\}}$, $x_{\{2,4\}}$.



Dégénérescence et risque de bouclage

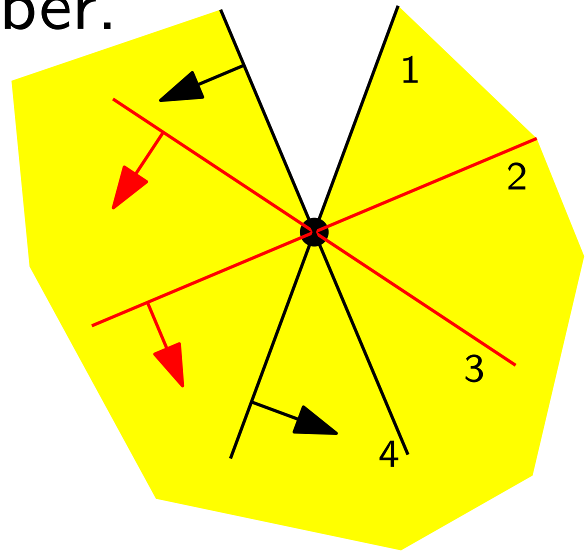
Dégénérescence: un sommet appartient à plus de n hyperplans.

Il n'est pas forcément acceptable de perturber.

Parmi les voisins de $x_{\{2,3\}}$ il y a

$x_{\{1,3\}}$, $x_{\{1,4\}}$, $x_{\{1,2\}}$, $x_{\{2,4\}}$.

Ce sont même ses *plus proches voisins*...



Dégénérescence et risque de bouclage

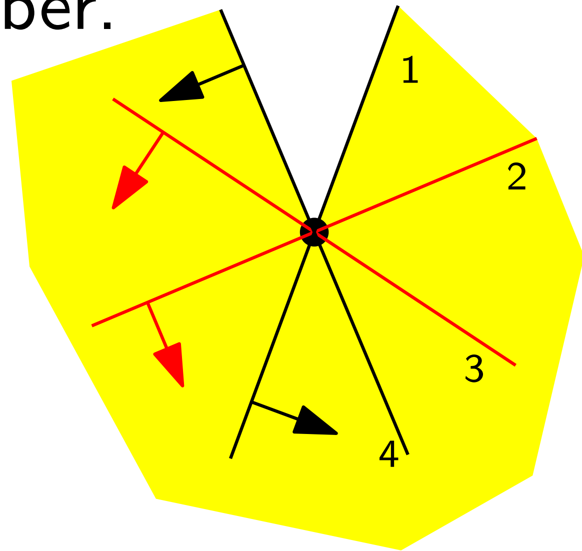
Dégénérescence: un sommet appartient à plus de n hyperplans.

Il n'est pas forcément acceptable de perturber.

Parmi les voisins de $x_{\{2,3\}}$ il y a

$x_{\{1,3\}}$, $x_{\{1,4\}}$, $x_{\{1,2\}}$, $x_{\{2,4\}}$.

Ce sont même ses *plus proches voisins*...



En dimension supérieure il peut être nécessaire
de changer plusieurs lignes pour trouver un vrai voisin dans P.

Il peut y avoir un nombre exponentiel de faux voisins.

Dégénérescence et risque de bouclage

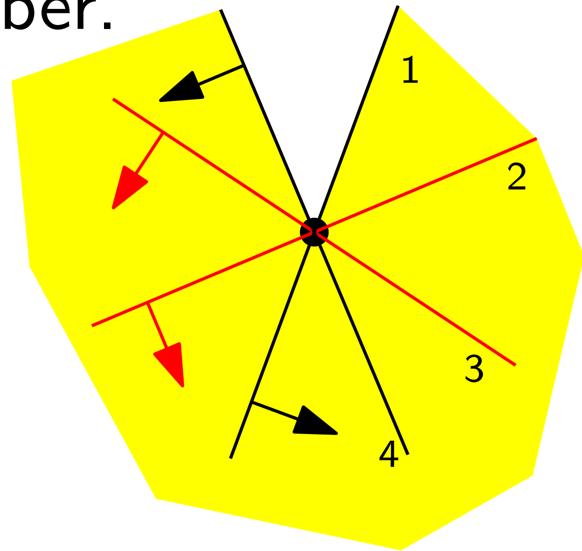
Dégénérescence: un sommet appartient à plus de n hyperplans.

Il n'est pas forcément acceptable de perturber.

Parmi les voisins de $x_{\{2,3\}}$ il y a

$x_{\{1,3\}}$, $x_{\{1,4\}}$, $x_{\{1,2\}}$, $x_{\{2,4\}}$.

Ce sont même ses *plus proches voisins*...



En dimension supérieure il peut être nécessaire
de changer plusieurs lignes pour trouver un vrai voisin dans P.

Il peut y avoir un nombre exponentiel de faux voisins.

Notre algorithme du simplexe prend un voisin le plus proche: il faut bien le choisir pour ne pas risquer de boucler indéfiniment en changeant I sans changer x_I: $x_{\{2,3\}} \longrightarrow x_{\{1,3\}} \longrightarrow x_{\{1,2\}} \longrightarrow x_{\{2,3\}}$.

Algorithme du simplexe, version finale

Soit x_I le sommet courant, solutions des n équations $(L_k x = b_k)_{k \in I}$

- Exprimer c dans la base $\{L_k\}_{k\in I}$: $c = \sum_{k\in I} y_k L_k$.

  Si $y_k \geq 0$ pour tout $k \in I$, on a trouvé l'optimum.

- Sinon on choisit le plus petit $i \in I$ tq $y_i < 0$ et on considère l'arête définie par les équations de $I' = I \setminus \{i\}$. Soit u son vecteur directeur tq $L_i u = -1$ (c'est une colonne de $(A_I)^{-1}$).

- Calculer la vitesse vers L_j quand on suit u: $v_j = L_j \cdot u$.

  si $v_j \leq 0$ on s'éloigne de l'hyperplan L_j: si $\forall j$, $\max = \infty$.

- Parmi les voisins sur l'arête, prendre celui qui est dans P, c'est celui qui est le plus proche parmi ceux du bon côté: le plus petit parmi les j tels que $v_j > 0$ qui minimisent $\frac{b_j - L_j x_I}{v_j}$.

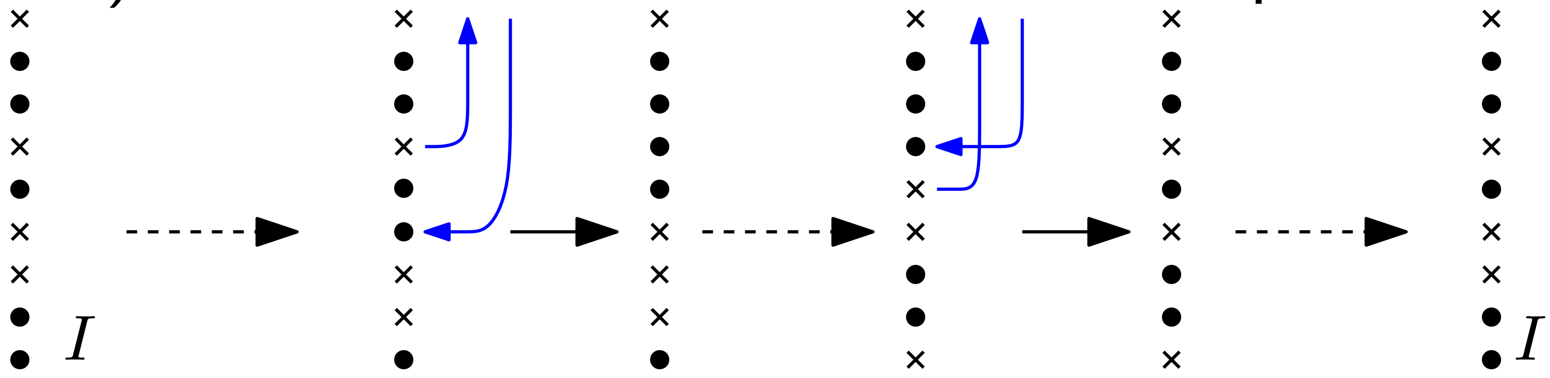
  faire $I = I \setminus \{i\} \cup \{j\}$, recalculer x_I et reprendre.

Algorithme du simplexe, preuve de terminaison

Si l'algo boucle c'est qu'on est revenu au même I, sans changer $c \cdot x_I$ ($c \cdot x_I$ croît). On observe l'évolution des indices présents dans I.

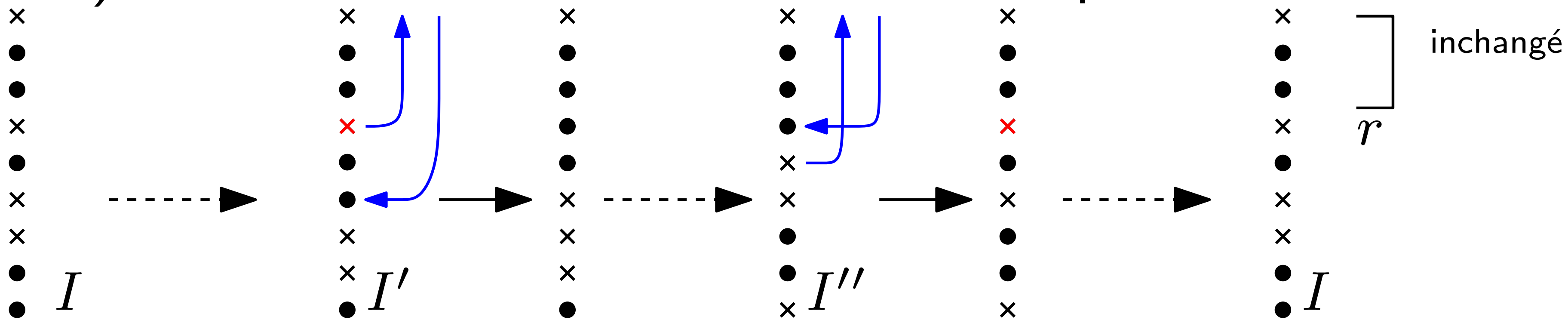
Algorithme du simplexe, preuve de terminaison

Si l'algo boucle c'est qu'on est revenu au même I, sans changer $c \cdot x_I$ ($c \cdot x_I$ croît). On observe l'évolution des indices présents dans I.



Algorithme du simplexe, preuve de terminaison

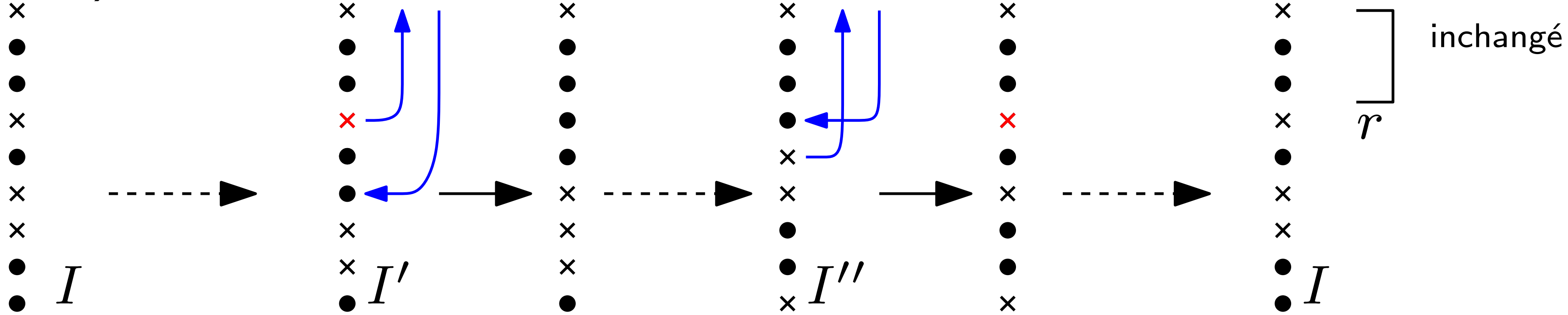
Si l'algo boucle c'est qu'on est revenu au même I, sans changer $c \cdot x_I$ ($c \cdot x_I$ croît). On observe l'évolution des indices présents dans I.



Soit r le plus grand indice qui est entre ou sort durant la boucle,
I' et I'' les ensembles d'indices à des temps d'entrée et de sortie de r.

Algorithme du simplexe, preuve de terminaison

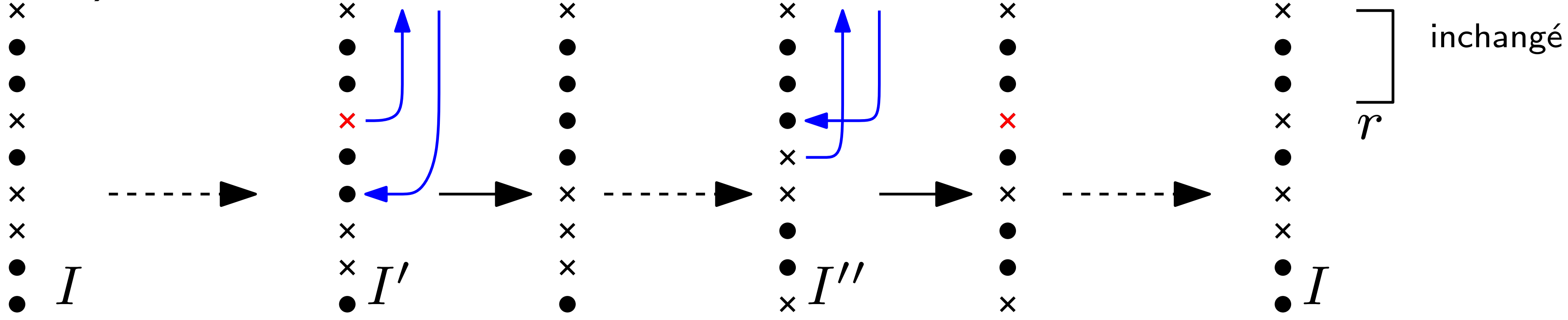
Si l'algo boucle c'est qu'on est revenu au même I, sans changer $c \cdot x_I$ ($c \cdot x_I$ croît). On observe l'évolution des indices présents dans I.



Soit r le plus grand indice qui est entre ou sort durant la boucle,
I' et I'' les ensembles d'indices à des temps d'entrée et de sortie de r.
Comme r sort de I': $c = \sum_{i \in I'} y_i L_i$, avec $y_i \geq 0$ pour $i < r$, et $y_r < 0$.

Algorithme du simplexe, preuve de terminaison

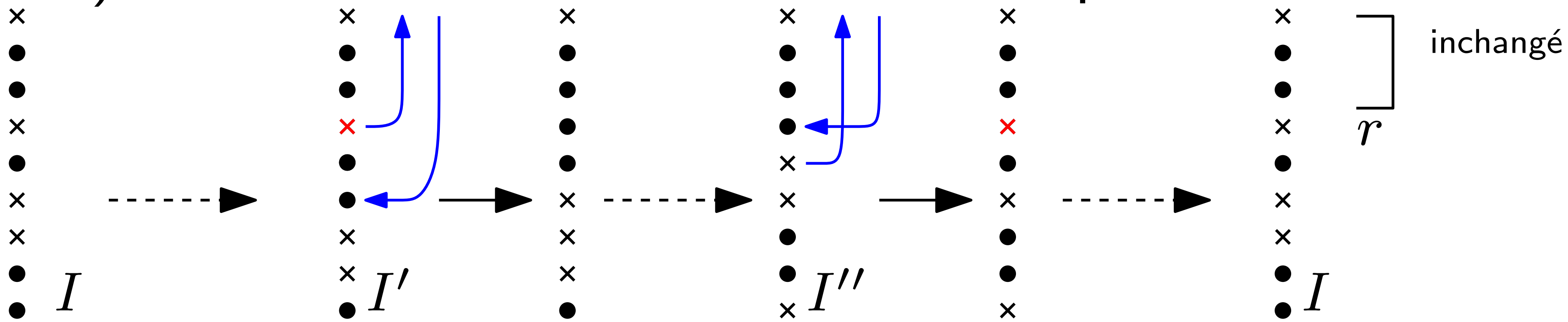
Si l'algo boucle c'est qu'on est revenu au même I, sans changer $c \cdot x_I$ ($c \cdot x_I$ croît). On observe l'évolution des indices présents dans I.



Soit r le plus grand indice qui est entre ou sort durant la boucle,
I' et I'' les ensembles d'indices à des temps d'entrée et de sortie de r.
Comme r sort de I': $c = \sum_{i \in I'} y_i L_i$, avec $y_i \geq 0$ pour $i < r$, et $y_r < 0$.
Soit u la direction de l'arête sélectionnée lors du traitement de I'':
alors $L_i u = 0$ pour $i \in I'' \setminus \{r\}$ et, comme r entre dans I'', on a pour $j < r$, $L_j u < 0$ dès que $b_j - L_j u = 0$, ce qui est le cas pour $j \in I' \setminus I''$.

Algorithme du simplexe, preuve de terminaison

Si l'algo boucle c'est qu'on est revenu au même I, sans changer $c \cdot x_I$ ($c \cdot x_I$ croît). On observe l'évolution des indices présents dans I.



Soit r le plus grand indice qui est entre ou sort durant la boucle,
I' et I'' les ensembles d'indices à des temps d'entrée et de sortie de r.
Comme r sort de I': $c = \sum_{i \in I'} y_i L_i$, avec $y_i \geq 0$ pour $i < r$, et $y_r < 0$.
Soit u la direction de l'arête sélectionnée lors du traitement de I'':
alors $L_i u = 0$ pour $i \in I'' \setminus \{r\}$ et, comme r entre dans I'', on a pour $j < r$, $L_j u < 0$ dès que $b_j - L_j u = 0$, ce qui est le cas pour $j \in I' \setminus I''$.
D'où $cu = \sum_{i \in I'} y_i L_i u = \sum_{i<r,\, i \in I'} y_i L_i u + y_r L_r u < 0$.

Ceci contredit le fait que suivant le u choisi on améliore l'objectif: ($cu > 0$).

Algorithme du simplexe, complexité

Puisque le nombre de voisins d'un sommet est $n(m-1)$ au plus, chaque étape a un coût polynomial.

Le nombre de sommets peut être exponentiel (jusqu'à $\binom{m}{n}$) en m et n, et il existe des exemples pour lesquels le simplexe visite effectivement un nombre exponentiel de sommets... (Worst case analysis).

On constate que cela ne se produit quasiment jamais en pratique...

On peut montrer qu'en moyenne sur des problèmes aléatoires l'algorithme est polynomial (Average case analysis).

Mieux on peut montrer que si on perturbe aléatoirement des données arbitraires, l'algorithme reste polynomial (Smooth analysis).

Programmation linéaire, algorithmes polynomiaux.

Le simplexe n'est pas polynomial mais presque...

De fait il existe des algorithmes polynomiaux pour la programmation linéaire: notamment les méthodes de l'ellipsoide et du point intérieur.

Attention toutefois: il y a des algorithmes polynomiaux si on cherche les optimaux à coordonnées dans $\mathbb{R}$.

On verra que, contrairement à ce qui se passe pour les problèmes de flots, le problème devient dur si on veut des solutions entières.

Cours 4: Programmation linéaire

- Position du problème
- Algorithme du simplexe générique
- Dualité.
- Dégénérescence et terminaison de l'algorithme

Retenir: - Il existe d'excellente boites noires pour résoudre les LP.
- Modélisation par LP (en PC). Thm Primal-Dual.