Programmation linéaire en nombres entiers

Alain Haït

ISAE – SUPAERO
Filière Sciences de la Décision

Novembre 2016

Plan du cours

Introduction

Programmation linéaire (PL)

- Variables de décision continues
- Problème polynomial
- Solveurs extrêmement performants

Programmation linéaire en nombres entiers (PLNE)

- Des variables entières ou binaires
- Problème NP-difficile dans le cas général
- Solveurs performants mais...

Introduction

Objectifs du cours

- Savoir modéliser un problème d'optimisation en PL ou PLNE.
- Connaître les principes de résolution des solveurs

Plus généralement :

- Trouver les bons modèles pour résoudre efficacement les problèmes combinatoires auxquels on est confronté
- Élaborer la stratégie de résolution la plus performante possible

Plan du cours

Programmation linéaire

À vous !

Plan du cours

Types de problèmes

- Programmation linéaire en nombres entiers

  $\max z = c^T x$
  Sous
  $Ax \leq b$
  $x_j \in \mathbb{N} \quad \forall j$

Types de problèmes

- Programmation en variables binaires

  $\max z = c^T x$
  Sous
  $Ax \leq b$
  $x_j \in \{0; 1\} \quad \forall j$

- Exemples : problème d'affectation, problème du sac-à-dos.

Types de problèmes

- Programmation en variables mixtes

  $\max z = c^T x$
  Sous
  $Ax \leq b$
  $x_j \in \{0; 1\} \quad \forall j \in [1; n_1]$
  $x_j \in \mathbb{R}^+ \quad \forall j \in [n_1 + 1; n]$

- Exemple : problème d'ordonnancement.

Programmation linéaire en nombres entiers

- Problèmes d'optimisation combinatoire
  - L'ensemble des solutions admissibles n'est plus continu.
  - On ne peut plus compter sur les sommets du polyèdre.
  - On ne peut généralement pas énumérer toutes les solutions.
  - On ne peut pas se contenter de la solution entière la plus proche de l'optimum continu.

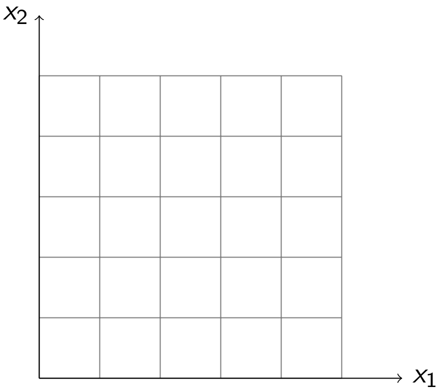
Exemple

$\max z = 19x_1 + 20x_2$

Sous

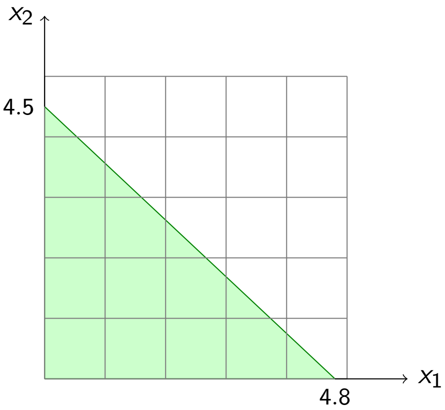
$15x_1 + 16x_2 \leq 72$

$x_1, x_2 \in \mathbb{N}$



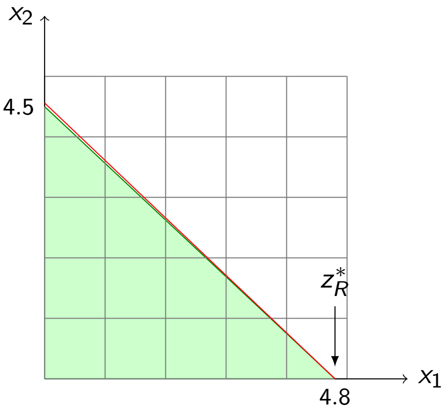
Exemple

max $z_R = 19x_1 + 20x_2$
Sous
$15x_1 + 16x_2 \leq 72$
$x_1, x_2 \in \mathbb{R}^+$



Exemple

$\max z_R = 19x_1 + 20x_2$
Sous
$15x_1 + 16x_2 \leq 72$
$x_1, x_2 \in \mathbb{R}^+$



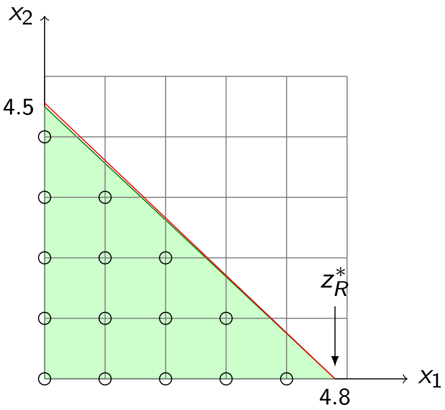
Exemple

$\max z = 19x_1 + 20x_2$

Sous

$15x_1 + 16x_2 \leq 72$

$x_1, x_2 \in \mathbb{N}$



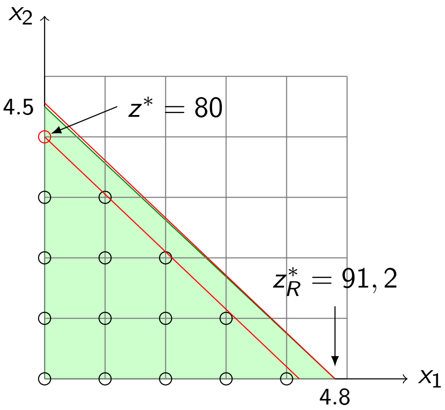
Exemple

$\max z = 19x_1 + 20x_2$

Sous

$15x_1 + 16x_2 \leq 72$

$x_1, x_2 \in \mathbb{N}$



Relaxations

Soit un problème d'optimisation (P) $\max z = f(x)$ sous la contrainte $x \in X \subseteq \mathbb{R}^n$.

Définition générale

Le problème (R) $\max z_R = \varphi(x)$ sous la contrainte $x \in T \subseteq \mathbb{R}^n$ est une relaxation de (P) si $X \subseteq T$ et $\varphi(x) \geq f(x) \quad \forall x \in X$.

Il est donc possible de relâcher le domaine (les contraintes), la fonction objectif, ou les deux.

Relaxations

Propriété :

- Si (R) est une relaxation de (P), alors $z_R^* \geq z^*$ (maximisation).

Qualité d'une relaxation

- Soient (R) et (R') deux relaxations de (P). La relaxation (R) est **meilleure** que (R') si $z_R^* \leq z_{R'}^*$

  C'est par exemple le cas si (R') est elle-même une relaxation de (R).

Relaxations

Relaxation linéaire d'un problème de programmation linéaire en nombres entiers :

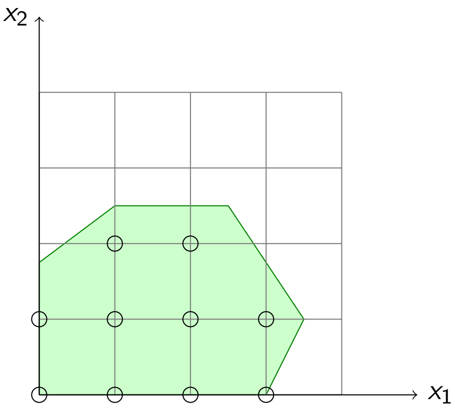
- Relaxation de la contrainte d'intégrité des variables
- Exploitation des caractéristiques du problème

Exemple

Problème (P) à résoudre :

$\max z = x_1 + 2x_2$
Sous

$$\begin{aligned} -3x_1 + 4x_2 &\leq 7 \\ 2x_2 &\leq 5 \\ 6x_1 + 4x_2 &\leq 25 \\ 2x_1 - x_2 &\leq 6 \\ x_1, x_2 &\in \mathbb{N} \end{aligned}$$

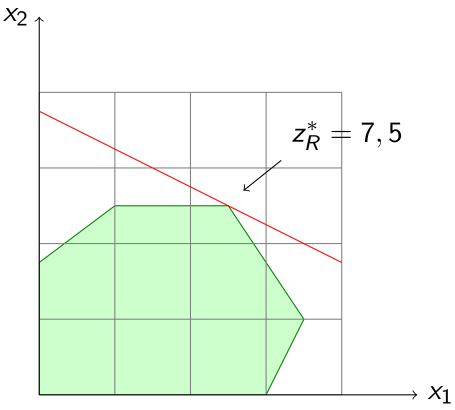


Exemple

(R) Relaxation linéaire de (P) :

max $z_R = x_1 + 2x_2$
Sous

$$
\begin{aligned}
-3x_1 + 4x_2 &\leq 7 \\
2x_2 &\leq 5 \\
6x_1 + 4x_2 &\leq 25 \\
2x_1 - x_2 &\leq 6 \\
x_1, x_2 &\in \mathbb{R}^+
\end{aligned}
$$

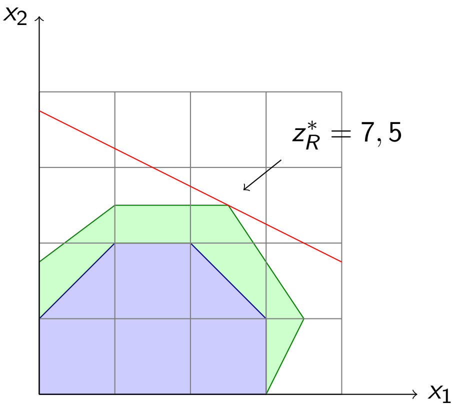


Exemple

(R') Autre relaxation linéaire de (P) :

$\max z_{R'} = x_1 + 2x_2$
Sous

$$\begin{aligned} -x_1 + x_2 &\leq 1 \\ x_2 &\leq 2 \\ x_1 + x_2 &\leq 4 \\ x_1 &\leq 3 \\ x_1, x_2 &\in \mathbb{R}^+ \end{aligned}$$

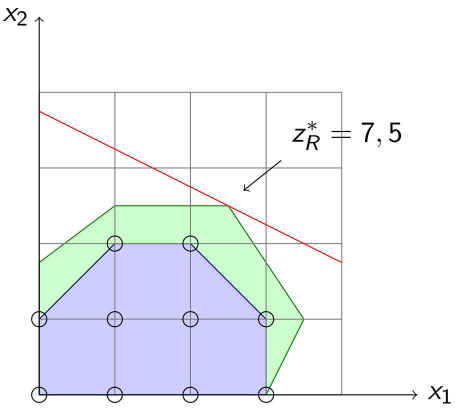


Exemple

(R') Autre relaxation linéaire de (P) :

$\max z_{R'} = x_1 + 2x_2$
Sous

$$\begin{aligned} -x_1 + x_2 &\leq 1 \\ x_2 &\leq 2 \\ x_1 + x_2 &\leq 4 \\ x_1 &\leq 3 \\ x_1, x_2 &\in \mathbb{R}^+ \end{aligned}$$

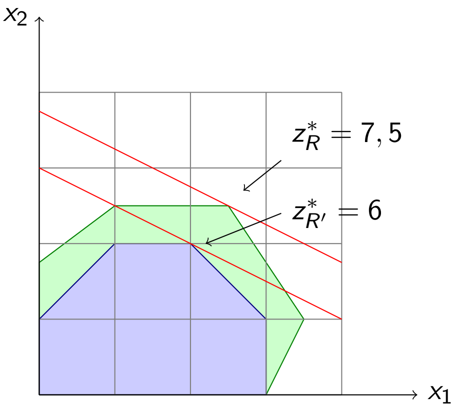


Exemple

(R') Autre relaxation linéaire de (P) :

$\max z_{R'} = x_1 + 2x_2$
Sous

$$\begin{aligned} -x_1 + x_2 &\leq 1 \\ x_2 &\leq 2 \\ x_1 + x_2 &\leq 4 \\ x_1 &\leq 3 \\ x_1, x_2 &\in \mathbb{R}^+ \end{aligned}$$



Résolution d'un problème de PLNE

Recherche d'une relaxation linéaire telle que $z_R^* = z^*$.

- La résolution de la relaxation linéaire de (P) sur l'enveloppe convexe des solutions admissibles entières de P donne la solution optimale.
- Mais trouver cette enveloppe convexe est en soi un problème combinatoire difficile.

Stratégies de résolution :

- Coupes : ajout de contraintes
- Recherche arborescente : résolution de sous-problèmes (séparation de X en plusieurs parties)
- Stratégies hybrides

Unimodularité

Classe de problèmes tels que, quelle que soit la valeur de b, la relaxation linéaire donne toujours une solution entière.

Propriété

Soit $B \in \mathbb{Z}^{m \times m}$ une matrice non singulière, alors $B^{-1}b$ est entier pour tout $b \in \mathbb{Z}^m$ si et seulement si $\text{Det}(B) = +\backslash -1$.

Une matrice $A \in \mathbb{Z}^{m \times p}$ de plein rang est unimodulaire si $\text{Det}(B) = +\backslash -1$ pour toute base B de A.

Unimodularité

Une matrice est totalement unimodulaire si chacune de ses sous-matrices a un déterminant égal à 0, 1 ou -1.

Corollaire

Si A est totalement unimodulaire, il existe une solution optimale de la relaxation continue qui résout le problème en nombres entiers.

Conditions suffisantes d'unimodularité :

- $a_{ij} \in \{-1; 0; 1\} \quad \forall i, j$
- $\sum_{i=1}^{m} |a_{ij}| \leq 2 \quad \forall j$
- Il existe une partition M_1, M_2 des indices des lignes telle que, si $\sum_{i=1}^{m} |a_{ij}| = 2$, alors $\sum_{i \in M_1} a_{ij} = \sum_{i \in M_2} a_{ij}$.

Plan du cours

Modélisation

Objectif :

Élaborer des modèles pertinents et dont la résolution est la plus performante possible.

Deux dimensions à équilibrer :

- Qualité de la relaxation linéaire
- Nombre de contraintes : compacité du modèle

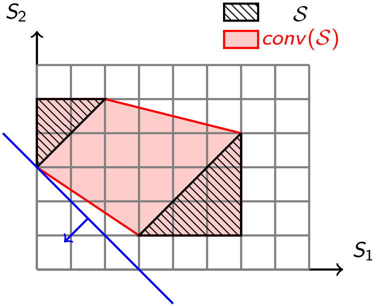
Dans cette partie du cours :

- Modélisation de disjonctions
- Modélisation de fonctions logiques
- Modélisation de non linéarités

Modélisation d'une disjonction

Exemple

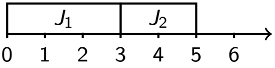
Un simple problème d'ordonnancement à deux tâches J_1 et J_2 de durées $p_1 = 3$ et $p_2 = 2$ sur une machine ne pouvant exécuter qu'une tâche à la fois. On doit déterminer les dates de débuts continues S_1 et S_2 sachant que chaque tâche doit démarrer dans une fenêtre de temps ($[0, 6]$ pour J_1 et $[1, 5]$ pour J_2). On cherche à minimiser $f(S) = S_1 + S_2 + p_1 + p_2$.



(PO) peut être résolu par la PL sur $conv(\mathcal{S})$. Impracticable en général

$$
\begin{aligned}
(PO)\ \text{Minimiser } & S_1 + S_2 + 5 \\
& S_1 \geq 0 \\
& S_2 \geq 1 \\
& S_1 \leq 6 \\
& S_2 \leq 5 \\
& S_2 \geq S_1 + 3 \lor S_1 \geq S_2 + 2
\end{aligned}
$$

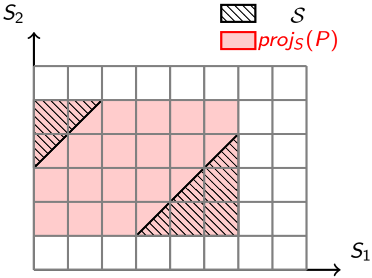
Solution optimale de valeur 8



Modélisation d'une disjonction par PLVM et qualité de la relaxation

- On introduit une variable binaire pour représenter les deux alternatives

$$
\begin{array}{rl}
(PLVM) \text{ Minimiser } & S_1 + S_2 + 5 \\
& S_1 \geq 0 \\
& S_2 \geq 1 \\
& S_1 \leq 6 \\
& S_2 \leq 5 \\
& S_2 - S_1 + x \geq 3 \\
& S_1 - S_2 + 7(1-x) \geq 2 \\
& x \in \{0,1\}
\end{array}
$$



La projection de P^E sur $\mathbb{R}^2$ donne bien $\mathcal{S}$

Valeur de relaxation = 6

Problème : $x = 0.5$ réalisable

$proj_S(P)$ **est éloigné de** $conv(\mathcal{S})$

Non linéarités

Dans certains cas on peut se ramener à un modèle linéaire.

- Min, Max, valeur absolue
- Fonctions linéaires par morceaux
- Multiplication de variables

Modélisation de fonctions élémentaires

- $f : [0, C]^2 \mapsto [0, C], z = f(x, y) =$ Minimiser (x, y)

$$\begin{cases} z \leq x \\ z \leq y \\ x \leq y + Cw \\ y \leq x + C(1 - w) \\ z \geq (1 - w)x + wy \\ w \in \{0, 1\} \\ 0 \leq x \leq C \\ 0 \leq y \leq C \end{cases}$$

La modélisation de f peut être simplifiée si le coefficient de z dans l'objectif est positif, dans le cadre d'une maximisation

Maximiser az

Sous les contraintes $\begin{cases} z \leq x \\ z \leq y \\ 0 \leq x \leq C \\ 0 \leq y \leq C \end{cases}$

avec $a > 0$

Modélisation de fonctions élémentaires

- $f : [a, b] \mapsto \mathbb{R}, z = f(x) = |x|$ avec $a < 0 < b$

$$\begin{cases} z = x^+ + x^- \\ x = x^+ - x^- \\ 0 \leq x^+ \leq by \\ 0 \leq x^- \leq |a|(1-y) \\ a \leq x \leq b \\ y \in \{0, 1\} \end{cases}$$

Multiplication de variables binaires

Produit de deux variables binaires x_i et x_j : $z = x_i.x_j$

$$\begin{aligned} z &\leq x_i \\ z &\leq x_j \\ z &\geq x_i + x_j - 1 \end{aligned}$$

Généralisation (Watters, 1967) : le produit de variables binaires $x_j, j \in J$ peut être remplacé par $z \in \{0, 1\}$ tel que :

$$\begin{aligned} \sum_{j \in J} x_j - z &\leq |J| - 1 \\ -\sum_{j \in J} x_j + |J|.z &\leq 0 \end{aligned}$$

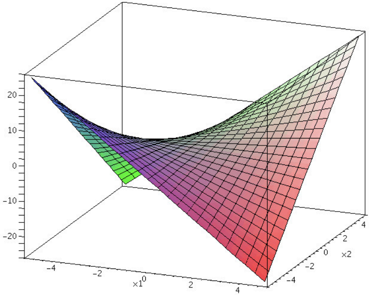
Multiplication de variables mixtes

Produit d'une variable binaire x et d'une variable réelle positive bornée $y \in [0; y_{sup}]$: z=x.y

$$z \leq y$$
$$z \leq x.y_{sup}$$
$$y - z \leq y_{sup}(1 - x)$$

Multiplication de deux variables réelles

Contrainte bilinéaire : $z = x.y$



Multiplication de deux variables réelles

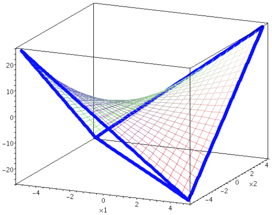
Relaxation de Mc Cormick (1976)

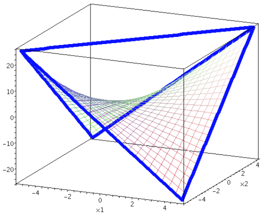
$$z \geq x_{inf}.y + y_{inf}.x - x_{inf}.y_{inf}$$

$$z \geq x_{sup}.y + y_{sup}.x - x_{sup}.y_{sup}$$

$$z \leq x_{sup}.y + y_{inf}.x - x_{sup}.y_{inf}$$

$$z \leq x_{inf}.y + y_{sup}.x - x_{inf}.y_{sup}$$





Modélisation de fonctions logiques

Rappels d'algèbre booléenne

- Une **proposition** est un énoncé qui est soit vrai, soit faux mais pas les deux à la fois (« tiers exclu »).
- La valeur de vérité de la proposition est dite **vraie** (**V**) si la proposition est vérifiée ; elle est **fausse** (**F**) dans le cas contraire.
- Soit P une proposition. La **négation** de P, notée $\neg P$ ou $\overline{P}$, est un opérateur logique qui prend la valeur opposée de P.

Rappels d'algèbre booléenne

On appelle **connecteur logique** un opérateur qui permet de former une nouvelle proposition à partir de deux propositions ou plus.

- **Conjonction** : la proposition « A et B », notée $A \wedge B$, est vraie si et seulement si A et B sont vraies.
- **Disjonction** : la proposition « A ou B », notée $A \vee B$, est fausse si et seulement si A et B sont fausses.
- **Implication** : la proposition « A implique B », notée $A \rightarrow B$, est fausse si et seulement si A est vraie et B est fausse.
- **Biconditionnelle** : notée $A \leftrightarrow B$, est vraie si et seulement si A et B ont la même valeur de vérité.

Rappels d'algèbre booléenne

Équivalence entre les propositions

Nom	Équivalence
Identité	$A \wedge \mathbf{V} \equiv A$ $A \vee \mathbf{F} \equiv A$
Domination	$A \vee \mathbf{V} \equiv \mathbf{V}$ $A \wedge \mathbf{F} \equiv \mathbf{F}$
Idempotence	$A \wedge A \equiv A$ $A \vee A \equiv A$
Double négation	$\neg(\neg A) \equiv A$
Commutativité	$A \wedge B \equiv B \wedge A$ $A \vee B \equiv B \vee A$
Associativité	$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$ $(A \vee B) \vee C \equiv A \vee (B \vee C)$

Rappels d'algèbre booléenne

Équivalence entre les propositions

Nom	Équivalence
Distributivité	$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
	$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
Loi de De Morgan	$\neg(A \wedge B) \equiv \neg A \vee \neg B$
	$\neg(A \vee B) \equiv \neg A \wedge \neg B$
Autres	$A \vee \neg A \equiv \mathbf{V}$
	$A \wedge \neg A \equiv \mathbf{F}$
	$A \rightarrow B \equiv \neg A \vee B$

Traduction en équations linéaires

Soit x_P une variable binaire associée à la proposition P. La variable x_P vaut 1 si la proposition P est vraie, 0 sinon.

Une **clause** est une disjonction inclusive de propositions. Pour traduire une expression logique en équations linéaires, on doit l'exprimer sous la forme d'une clause ou d'une conjonction de clauses.

Traduction en équations linéaires

Expression	Contraintes linéaires
A vrai	$x_A \geq 1$
$\neg A$ vrai	$1 - x_A \geq 1$
$A \wedge B$ vrai	$x_A \geq 1$ $x_B \geq 1$
$A \vee B$ vrai	$x_A + x_B \geq 1$
$A \rightarrow B$ vrai	$x_A \leq x_B$

Traduction en équations linéaires

Expression	Contraintes linéaires
$P \leftrightarrow A$	$x_P = x_A$
$P \leftrightarrow \neg A$	$x_P = 1 - x_A$
$P \leftrightarrow A \wedge B$	$x_P \leq x_A$ $x_P \leq x_B$ $x_P \geq x_A + x_B - 1$
$P \leftrightarrow A \vee B$	$x_P \geq x_A$ $x_P \geq x_B$ $x_P \leq x_A + x_B$
$P \leftrightarrow A \rightarrow B$	$x_P \geq 1 - x_A$ $x_P \geq x_B$ $x_P \leq 1 - x_A + x_B$

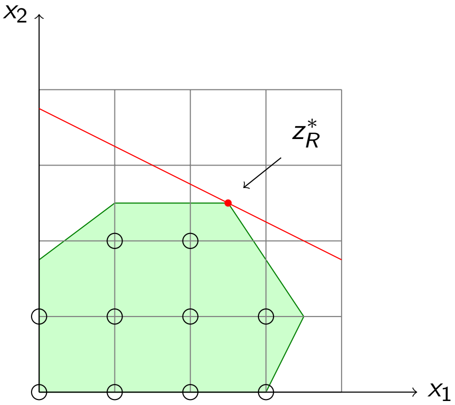
Problèmes classiques

- Gestion de projets : resource-constrained Project Scheduling Problem (RCPSP)
- Planification de production : Lot Sizing
- Production d'énergie : Unit Commitment
- Tournées : Traveling Salesman Problem (TSP)
- Tournées : Vehicle Routing Problem (VRP)
- etc.

Plan du cours

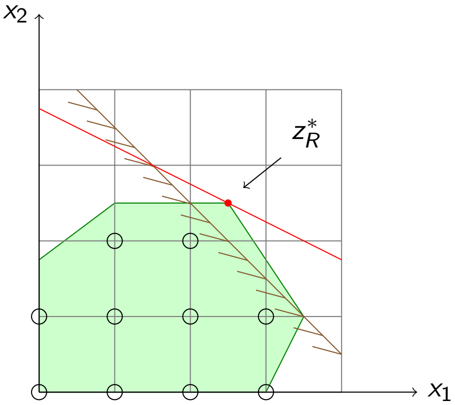
Coupes

Principe : ajout de contraintes à la relaxation linéaire (R) pour que l'optimum de (P) apparaisse en un sommet de (R).



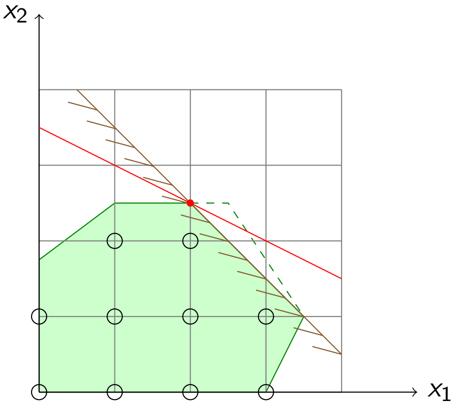
Coupes

Principe : ajout de contraintes à la relaxation linéaire (R) pour que l'optimum de (P) apparaisse en un sommet de (R).



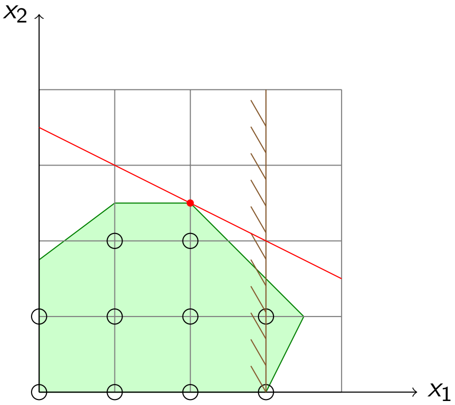
Coupes

Principe : ajout de contraintes à la relaxation linéaire (R) pour que l'optimum de (P) apparaisse en un sommet de (R).



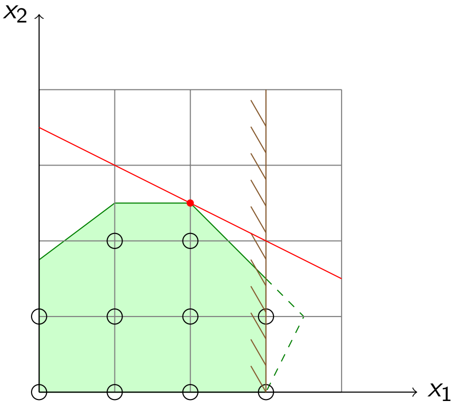
Coupes

Principe : ajout de contraintes à la relaxation linéaire (R) pour que l'optimum de (P) apparaisse en un sommet de (R).



Coupes

Principe : ajout de contraintes à la relaxation linéaire (R) pour que l'optimum de (P) apparaisse en un sommet de (R).



Coupes

Principe :

- L'ensemble des solutions de (P) est conservé
- La solution optimale courante (R) est exclue

Détermination de coupes :

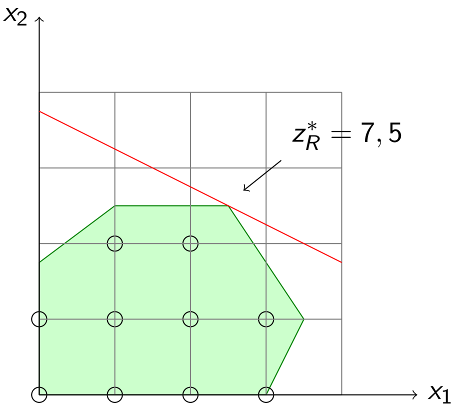
1. Coupes d'arrondi à partir de l'objectif et des contraintes
2. Coupes spécifiques à un problème
3. Coupes généralisées : coupes de Gomory

Coupes d'arrondi

Problème (P) :

$\max z = x_1 + 2x_2$
Sous

$$\begin{aligned} -3x_1 + 4x_2 &\leq 7 \\ 2x_2 &\leq 5 \\ 6x_1 + 4x_2 &\leq 25 \\ 2x_1 - x_2 &\leq 6 \\ x_1, x_2 &\in \mathbb{N} \end{aligned}$$

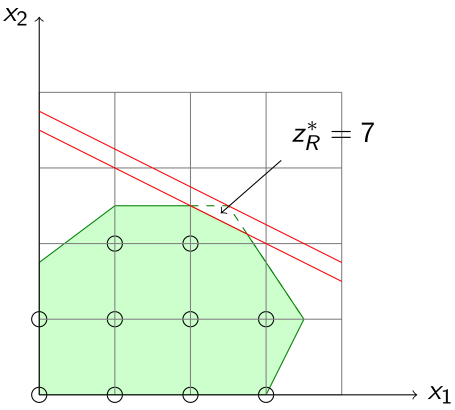


Coupes d'arrondi

Ajout d'une contrainte :

$\max z = x_1 + 2x_2$
Sous

$$-3x_1 + 4x_2 \leq 7$$
$$2x_2 \leq 5$$
$$6x_1 + 4x_2 \leq 25$$
$$2x_1 - x_2 \leq 6$$
$$x_1 + 2x_2 \leq \lfloor 7,5 \rfloor$$
$$x_1, x_2 \in \mathbb{N}$$

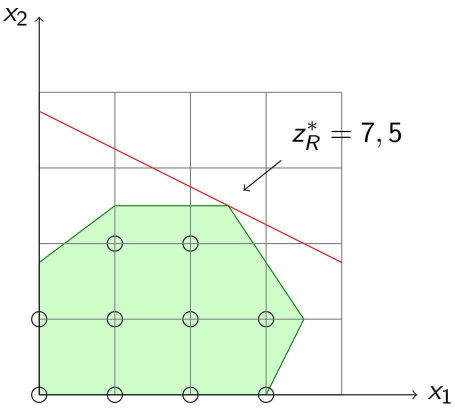


Coupes d'arrondi

Problème (P) :

$\max z = x_1 + 2x_2$
Sous

$$
\begin{aligned}
-3x_1 + 4x_2 &\leq 7 \\
2x_2 &\leq 5 \\
6x_1 + 4x_2 &\leq 25 \\
2x_1 - x_2 &\leq 6 \\
x_1, x_2 &\in \mathbb{N}
\end{aligned}
$$

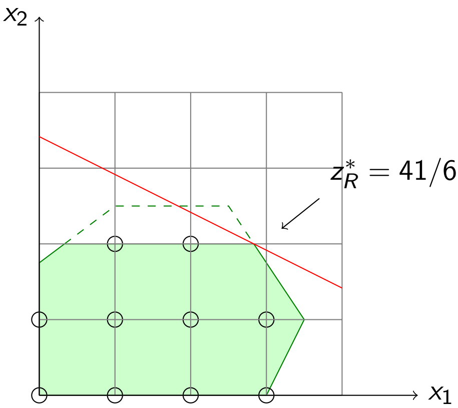


Coupes d'arrondi

Contrainte (2) :

$\max z = x_1 + 2x_2$
Sous

$$-3x_1 + 4x_2 \leq 7$$
$$x_2 \leq \lfloor 5/2 \rfloor$$
$$6x_1 + 4x_2 \leq 25$$
$$2x_1 - x_2 \leq 6$$
$$x_1, x_2 \in \mathbb{N}$$

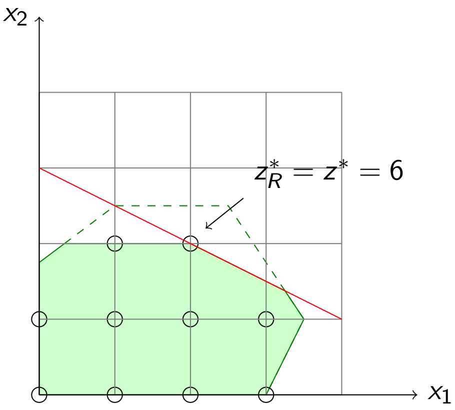


Coupes d'arrondi

Fonction objectif :

$\max z = x_1 + 2x_2$
Sous

$$-3x_1 + 4x_2 \leq 7$$
$$x_2 \leq 2$$
$$6x_1 + 4x_2 \leq 25$$
$$2x_1 - x_2 \leq 6$$
$$x_1 + 2x_2 \leq \lfloor 41/6 \rfloor$$
$$x_1, x_2 \in \mathbb{N}$$

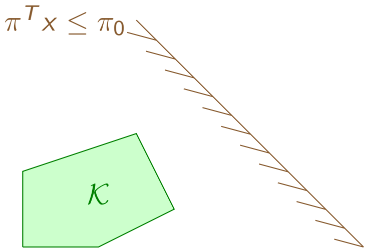


Définitions

Soit $\mathcal{K} = \{x \in \mathbb{R}^n : Ax \leq b\}$ un polyèdre de $\mathbb{R}^n$.

- Une inégalité $\pi^T x \leq \pi_0$ est dite **valide** pour $\mathcal{K}$ si elle est satisfaite pour tout x appartenant à $\mathcal{K}$:

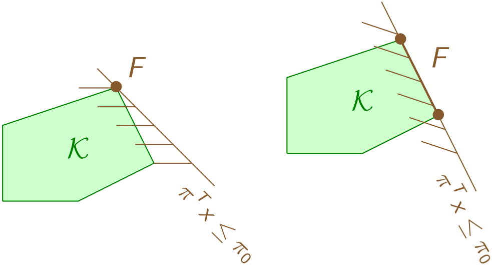
$$\mathcal{K} \subseteq \{x \in \mathbb{R}^n : \pi^T x \leq \pi_0\}$$



Définitions

Facettes d'un polyèdre

- Soit $\pi^T x \leq \pi_0$ une inégalité valide pour le polyèdre $\mathcal{K}$ de $\mathbb{R}^n$.
- $F = \{x \in \mathcal{K} : \pi^T x = \pi_0\}$ est une **face** de $\mathcal{K}$ si $F \neq \varnothing$.
- $F = \{x \in \mathcal{K} : \pi^T x = \pi_0\}$ est une **facette** de $\mathcal{K}$ si $F \neq \varnothing$ et F est de dimension $n - 1$.



Définitions

Enveloppe convexe

- Soit $S = \{x^1, x^2, \ldots, x^k\}$ ensemble de points de $\mathbb{R}^n$.
- $\text{CONV}(S) = \{x \in \mathbb{R}^n : \exists \lambda_i \in \mathbb{R}_+$ tels que $\sum_{i=1}^k \lambda_i = 1$ et $x = \sum_{i=1}^k \lambda_i . x^i\}$.

Hyperplan de séparation

- Soit $S = \{x^1, x^2, \ldots, x^k\}$ ensemble de points de $\mathbb{R}^n$.
- Soit $x^0 \in \mathbb{R}^n \backslash \text{CONV}(S)$.
- Alors il existe $\pi \in \mathbb{R}^n$ et $\pi_0 \in \mathbb{R}$ tels que

$$\begin{aligned} \pi^T x &\leq \pi_0 \quad \forall x \in \text{CONV}(S) \\ \pi^T x^0 &> \pi_0 \end{aligned}$$

Ajout de coupes en PLNE

Détermination d'hyperplans de séparation entre

- La solution de la relaxation linéaire du problème, si elle n'est pas entière
- L'enveloppe convexe des solutions entières du problème.

L'enjeu est de trouver les coupes les plus performantes : celles qui permettent de se rapprocher le plus vite de la solution optimale du problème en nombres entiers.

Coupes spécifiques

Coupes adaptées à une catégorie de problèmes

- Coupes de cliques
- Coupes de couverture (sac-à-dos)
- *etc.*

Coupes de cliques

- Clique : ensemble de variables $C = \{x_1, \ldots, x_n\}$ tel qu'il existe une contrainte d'exclusion entre chaque couple de variables de C :

$$x_{j1} + x_{j2} \leq 1 \quad \forall (x_{j1}, x_{j2}) \in C^2, j1 < j2$$

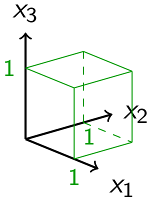
- Pour toute clique l'inégalité suivante est valide :

$$\sum_{j=1,\ldots,n} x_j \leq 1$$

- Clique maximale

Coupes de cliques

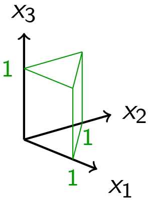
Exemple à trois variables x_1, x_2, x_3



Domaine des variables

Coupes de cliques

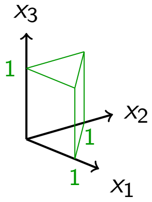
Exemple à trois variables x_1, x_2, x_3



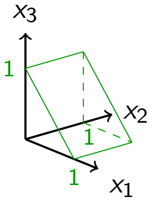
$x_1 + x_2 \leq 1$

Coupes de cliques

Exemple à trois variables x_1, x_2, x_3



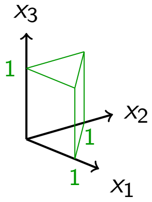
$x_1 + x_2 \leq 1$



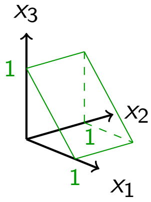
$x_1 + x_3 \leq 1$

Coupes de cliques

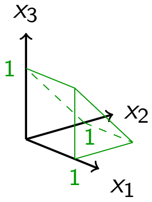
Exemple à trois variables x_1, x_2, x_3



$x_1 + x_2 \leq 1$



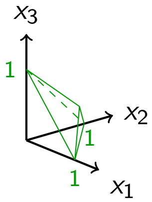
$x_1 + x_3 \leq 1$



$x_2 + x_3 \leq 1$

Coupes de cliques

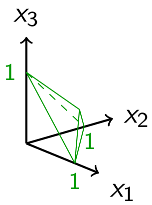
Exemple à trois variables x_1, x_2, x_3



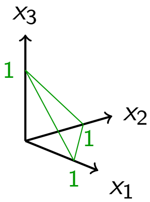
Trois contraintes d'exclusion

Coupes de cliques

Exemple à trois variables x_1, x_2, x_3



Trois contraintes d'exclusion



Coupe de clique $x_1 + x_2 + x_3 \leq 1$

Coupes de couverture

On considère le problème de sac à dos :

$\max c^T x$
sous les contraintes :
$a.x \leq b$
$x \in \{0;1\}^n$
où $c \in \mathbb{Z}_+^n, a^T \in \mathbb{Z}_+^n, b \in \mathbb{Z}_+$.

Coupes de couverture

- Soit $C \subset J = \{1, \ldots, n\}$. On dit que C est un **ensemble dépendant** si $\sum_{j \in C} a_j > b$.
- Si C est un ensemble dépendant, alors l'inégalité :

$$\sum_{j \in C} x_j \leq |C| - 1,$$

  appelée coupe de couverture, est valide.
- Sous certaines conditions, les coupes de couverture peuvent correspondre à des facettes du problème du sac à dos.

Génération de coupes

Principe :

- Coupes issues de la structure du programme linéaire
- Excluent la solution courante
- Sont valides pour l'ensemble des solutions entières

Génération de coupes

Inégalités de Chvatál-Gomory

- Problème (P) : $\max z = c^T x$ sous $Ax \leq b$ et $x \in \mathbb{N}^n$
- Toutes les lignes $A_i.x \leq b_i \quad i = 1, \ldots, m$ sont des inégalités valides pour l'enveloppe convexe des solutions admissibles de (P).
- Soit $y \in \mathbb{R}^m_+$
- Alors $\pi^T x \leq \pi_0$, avec

$$\pi^T = \left\lfloor \sum_{i=1}^{m} y_i.A_i \right\rfloor \in \mathbb{Z}^n$$
$$\pi_0 = \left\lfloor \sum_{i=1}^{m} y_i.b_i \right\rfloor \in \mathbb{Z}$$

est une inégalité valide pour l'enveloppe convexe des solutions de (P).

Génération de coupes

Propriété

Toute inégalité valide pour l'enveloppe convexe des solutions admissibles de P peut être obtenue en générant un nombre fini d'inégalités de Chvatál-Gomory.

Coupes fractionnaires de Gomory

Problème mis sous forme simpliciale dans la base optimale $\hat{B}$:

$x_{\hat{B}} = \hat{B}^{-1}b - \hat{B}^{-1}\hat{N}.x_{\hat{N}} = \hat{B}^{-1}b - \sum_{j\in\hat{N}} \hat{a}_{ij}x_j$

$z = z^* + \sum_{j\in\hat{N}} \hat{c}_j x_j$

Considérons la contrainte i

- Si $\hat{b}_i = (\hat{B}^{-1}b)_i$ n'est pas entier, on introduit une coupe.
- Soit f_i la partie fractionnaire de $\hat{b}_i$ et f_{ij} la partie fractionnaire de $\hat{a}_{ij}$.
- $\sum_{j\in\hat{N}} f_{ij}x_j \geq f_i$ est une coupe associée à la contrainte i.

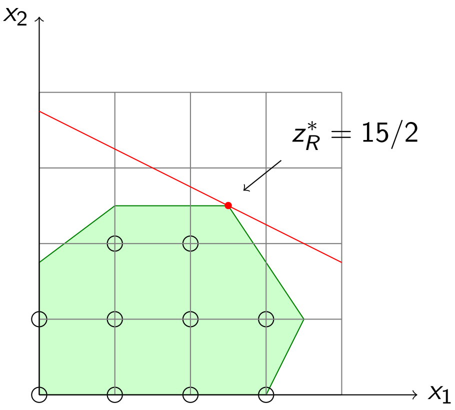
Remarque : Il n'est pas obligatoire d'atteindre la base optimale de la relaxation.

Exemple

Problème (P) :

$\max z = x_1 + 2x_2$
Sous

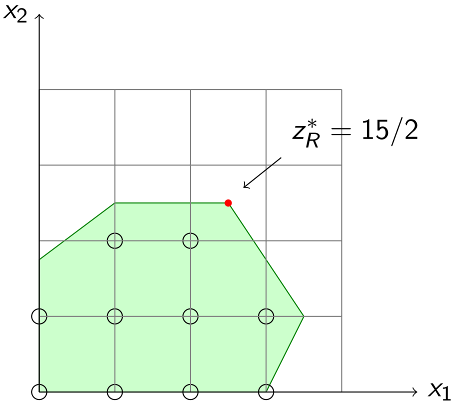
$$-3x_1 + 4x_2 \leq 7$$
$$2x_2 \leq 5$$
$$6x_1 + 4x_2 \leq 25$$
$$2x_1 - x_2 \leq 6$$
$$x_1, x_2 \in \mathbb{N}$$



Exemple

Tableau final de (R) :

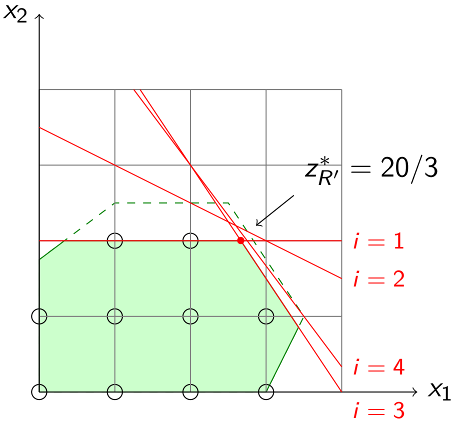
		x_4	x_5
x_1	5/2	-1/2	0
x_2	5/2	1/3	-1/6
x_3	9/2	3	-1/2
x_6	7/2	-7/6	1/3
z_R	15/2	-7/6	-1/6



Exemple

Tableau final de (R) :

		x_4	x_5
x_1	5/2	-1/2	0
x_2	5/2	1/3	-1/6
x_3	9/2	3	-1/2
x_6	7/2	-7/6	1/3
z_R	15/2	-7/6	-1/6



Plan du cours

Méthodes par séparation et évaluation

Cas général : maximisation de $f(x)$ avec $x \in X$.

- **Séparation** de l'ensemble des solutions admissibles X en plusieurs sous-ensembles.
- Dans chaque sous-ensemble l'**évaluation** de f donne une borne supérieure de l'optimum sur ce sous-ensemble.
- **Comparaison** des résultats sur les différents sous-ensembles pour ne pas développer toute l'arborescence.

Algorithme

- Parcours de l'arborescence jusqu'à obtention de l'optimum ou conclusion sur son existence.

Séparation

- Le premier ensemble à séparer est X
- X est séparé en $X^1, X^2, \ldots, X^K$ tels que
  - On ne perd pas de solution
    $X^k \subset X \quad k = 1, \ldots, K$
  - On ne crée pas de solution
    $X^1 \cup X^2 \cup \ldots \cup X^K = X$

Questions : comment séparer ? Combien de sous-ensembles ?

Évaluation

On sait évaluer un sous-ensemble X^i si on connaît une relaxation du problème :

- L'évaluation $\varphi(X^i)$ est telle que $\varphi(X^i) \geq f(x) \quad \forall x \in X^i$
- L'évaluation est **exacte** si on connaît $\hat{x}^i \in X^i$ admissible tel que $\varphi(X^i) = f(\hat{x}^i)$
- L'évaluation φ' est meilleure que φ si $\varphi'(X^i) \leq \varphi(X^i)$
- Une solution admissible de X^i donne une borne inférieure sur ce sous-ensemble et tous ses parents

Questions : quelle(s) relaxation(s) choisir ?

Sondage

On sonde les sous-ensembles pour conclure sur la présence de l'optimum global :

- **Admissibilité** : s'il n'y a plus de solution admissible dans le sous-ensemble
- **Résolution** : si on a trouvé une évaluation exacte du sous-ensemble
- **Optimalité** : si on connaît une solution admissible x^k telle que

$$\varphi(X^i) \leq f(x^k)$$

Si on ne peut pas conclure sur un sous-ensemble, il doit être séparé.

Stratégies de recherche

Parcours de l'arborescence :

- En largeur : choix du plus ancien nœud apparu.
- En profondeur : choix du dernier nœud apparu.
- Meilleur d'abord : choix du nœud le plus prometteur suivant un critère donné.
- Stratégies hybrides.

Application à la PLNE

Variables entières ou binaires

- Évaluation : relaxation linéaire
- Séparation : plans sécants
- Parcours de l'arborescence
- Association avec des coupes : Branch and Cut

Plan du cours

Briques de base des solveurs

- Prétraitement
- Ajout de coupes
- Stratégies de branchement
- Recherche de solutions admissibles

Prétraitement

Sur le modèle :

- Suppression d'éléments redondants
- Bornes des variables

Algorithmique :

- Détection de sous-structures : graphes, cliques...
- Propagation de contraintes

Branchement

Branchement fort

- Simulation du branchement sur les variables candidates
- Sélection de la meilleure amélioration de borne

Branchement pseudocoût

- Historique des branchements antérieurs

Branchement fiable

- Pseudocoût jusqu'à une certaine limite
- Branchement fort au-delà.

Autres caractéristiques

- Ajustement automatique des paramètres
- Tirages aléatoires
- Parallélisme
- Détection des infaisabilités
- Solutions multiples
- Pilotage de la recherche

A retenir

Complexité de la recherche arborescente

- Par définition, il existe dans l'arborescence un chemin polynomial jusqu'à la solution optimale. En revanche, à moins que $\mathcal{P} = \mathcal{NP}$, le chemin suivi dans le pire cas est **exponentiel**.
- Les algorithmes de séparation et évaluation (*branch and bound*) explorent l'arborescence de manière **heuristique**. Cette recherche peut être malchanceuse !

Sources et compléments I

Michel Minoux.
Programmation mathématique.
Lavoisier, 2008, 2e édition.

Dominique de Werra, Thomas Liebling, Jean-François Hêche.
Recherche opérationnelle pour ingénieurs.
Presses polytechniques et universitaires romandes, 2003.

Stephen Bradley, Arnoldo Hax, Thomas Magnanti
Applied mathematical programming.
MIT Press, Addison-Wesley, 1977.
http ://web.mit.edu/15.053/www/

Laurence Wosley
Integer programming.
Wiley -Interscience, 1998.

Sources et compléments II

- **François Soumis**
  Modélisation en recherche opérationnelle.
  Support de cours École Polytechnique de Montréal.
- **Michel Gamache**
  Recherche opérationnelle minière.
  Support de cours École Polytechnique de Montréal.
- **Andrea Lodi**
  Problem-solving by Mixed-Integer Programming.
  Présentation à ROADEF 2014.