La programmation mathématique en nombres entiers

Modeliser

Stanislas Francfort

Orange Labs

5 Juin 2017

But du cours

- Différence Programmation Linéaire (PL) et Programmation Linéaire en Nombres Entiers (PLNE)
- Etre capable de modéliser un PLNE
- Connaitre les méthodes avancées de résolutions de PLNE
- Savoir mettre en oeuvre ces méthodes

Plan du cours

- Introduction
- Modélisation
- Relaxations
- Polytopes à solutions entières
- Méthodes d'énumérations
- Algorithmes de coupes
- Reformulation / Décomposition/ génération de colonnes
- Approche polyèdrale
- Exemples d'implémentations, et résolution

Définitions et historique

- la Recherche Opérationnelle (RO) : l'ensemble des domaines scientifiques, des outils et des problèmes touchant aux questions d'ordre décisionnel (dit aussi stratégique) ou d'optimisation de systèmes complexes.
- Les problèmes sont rendus complexes par
  - la dimension du problème peut être importante
  - la structure peut-être par combinatoire, concurrentielle, stochastique
- des exemples :
  - la recherche d'un itinéraire sur une carte
  - l'ordonnancement des taches a accomplir en usine
  - la décision stratégique d'investissement d'une entreprise sur un marché concurrentiel

Historique

- les débuts : les "jeux mathématiques" des mathématiciens depuis le XVIème siècle (Blaise Pascal, Euler,...)
- l'utilisation dans la société : la seconde guerre mondiale.
  - l'organisation, par l'armée des Alliés, du débarquement en Normandie.
  - gestion au mieux de l'implantation rapide de radars
  - d'acheminement de troupes
  - la gestion de leur alimentation
  - des contacts entre unité etc.
- L'armée alliée fit appel a des mathématiciens (et quelques premiers informaticiens)

Où a lieu la RO

- **Pour l'entreprise :** un outil informatique pour aider à prendre des décisions
- Les grands groupes industriels ont quasiment toujours un département R0
- dans des départements de R&D d'entreprise
- dans des laboratoires académiques
- dans des PME
- pour des groupes de consultants

Les consultants R0

- Les consultants en R0 sont demandés par les entreprises pour répondre a des besoins précis ou ponctuels.
- Ils sont parfois universitaires, répandu dans le monde anglo-saxon, plus rare en France.
- Il existe de plus en plus de petites sociétés de service (SSII, consulting, conseil stratégique,...). Spécialisés sur un type de problèmes (transport, gestion, planification, etc) ou sur un logiciel de résolution : parmi elles en France Artelys, Eurodecision, Dynasys,... 0n peut les diviser en deux grandes catégories :
    - les entreprises de décisions stratégiques
    - les entreprises de conception logicielle R0

Les outils

- nombreux outils logiciels commericaux ou libres
  - Solveur linéaire (cplex, LP de Coin-OR, xpress, glpk, gurobi, excel...)
  - Solveur entier (cplex, xpress, glpk, gurobi,cbc...)
  - Solveur quadratique (cplex, xpress)
  - Solveur continu, solveur pour programme semi-défini,...
- et des API, des solutions
- **pour la programmation en nombres entiers, python/pyomo/cbc julia/cbc fonctionnent bien et sont open-source**

Les grandes problématiques de la RO

- network design (conception de réseaux de télécommunication)
- transport
- localisation (plant location)
- micro-électronique (VLSI design)
- bio-informatique, applications médicales
- productique (job-shop, flow-shop,...)
- ordonnancement (domaine a la limite problématique / problème...)
- optimisation financière
- chaîne logistique (supply chain) et création de lots (lot-sizing)
- optimisation “durable”

La pratique de la R0

- Découvrir le problème
- Modéliser
- Valider le modèle
- Evaluer la complexité de l'état de l'art
- Choisir un outil (s'il existe)
- Etude scientifique et algorithmique
- Expérimentation et retour d'expérience
- Intégration et suite logicielle

Définitions

Un Programme Mathématique, noté PM, est un problème d'optimisation sous contrainte $(\mathcal{P})$:

$$\begin{aligned} \text{Maximiser } & f(x) \\ & \text{sous les contraintes} \\ & g_i(x) \leq 0 \forall i = 1, \ldots, m \\ & x \in S. \end{aligned}$$

où

- $S \subseteq \mathbb{R}^n$ et x un vecteur appelé variable (ce sont les inconnues),
- la fonction $f : S \to \mathbb{R}^n$ est appelée fonction objective,
- les fonctions $g_i : S \to \mathbb{R}^n, i = 1, \ldots, m$, sont des inégalités appelées contraintes.

Définitions

- Un PM peut ètre une maximisation ou une minimisation (poser $f' = -f$).
- les contraintes peuvent être ≤ 0, $\neq 0$ ou $= 0$.
- Si $\overline{x}$ vérifie les contraintes alors on l'appelle solution réalisable
- L'ensemble des solutions d'un PM forme un domaine de définition. Il peut être vide
- Il peut exister des solutions $x^{\star}$ qui maximisent la fonction $f(x)$

- si S est continu, on parle de PM continu
- si S est discret (c'est-a-dire isomorphe a une sous-partie de $\mathbb{N}$), on parle de PM discret (PMD), ou PM entier
- si certaines composantes du vecteur x sont entières et d'autres continues, on parle de PM mixte

Définitions

Dans le cas des PMD, on peut noter alors :

$$\begin{aligned} \text{Maximiser } & f(x) \\ & \text{sous les contraintes} \\ & g_i(x) \leq 0 \forall i = 1, \dots, m \\ & x \in \mathbb{Z}. \end{aligned}$$

On appelle alors $x \in \mathbb{Z}$ la contrainte d'intégrité (ou d'entiéreté en Belgique ou d'intégralité au Québec).
On appelle **relaxation** le fait de "relâcher" (supprimer) une contrainte du problème.
On appelle relaxation continue le fait de "relâcher" les contraintes d'intégrité du problème.

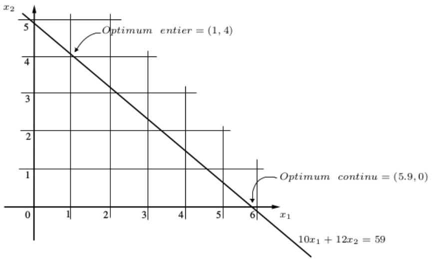
Difficulté de la PMD

- La programmation mathématique discrète est un domaine plutôt récent.
- En fait, on sait parfois bien résoudre des PMD particuliers
- **mais on ne connait pas de méthode générale pour résoudre n'importe quel PMD**
- Dans ce cours nous verrons quelques cas particuliers
  - problèmes que l'on sait résoudre
  - problèmes que l'on ne sait pas résoudre
  - méthodes efficaces dans certains cas
  - méthodes permettant de se ramener à des cas plus faciles

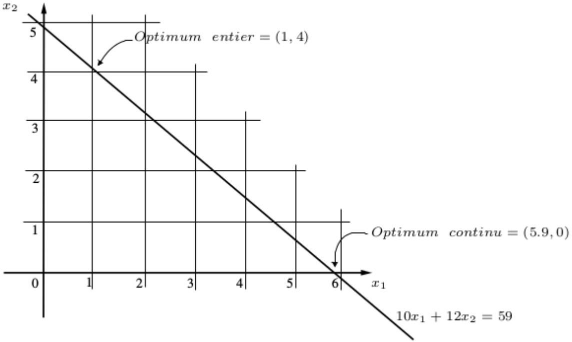
Difficulté de la PMD

Il ne suffit pas “d’arrondir” la solution vers sa relaxation continue.

$$
\begin{aligned}
&\text{Maximiser } 10x_1 + 11x_2\\
&\quad 10x_1 + 12x_2 \leq 59\\
&\quad x_1 \text{ et } x_2 \geq 0\\
&\quad x_1, x_2 \text{ entiers}
\end{aligned}
$$



Difficulté de la PMD



- l'optimum de la relaxation continue : 59
- l'optimum entier : 54 seulement.
- on remarque la différence de structure et de position des deux points optimum

Difficulté de la PMD

- résoudre un (PMD) donné est au moins aussi difficile que résoudre sa relaxation continue.
- Souvent, il l'est bien davantage
- certains problèmes ont des relaxations continues polynomiales et des versions discrètes NP-difficiles (problèmes linéaires).
- Inversement, si l'on ne sait pas résoudre efficacement la version continue d'un programme, il est quasi impensable de résoudre sa version discrète.

Programme linéaire

- Si la fonction objective f et les fonctions contraintes g sont toutes linéaires : on parle de Programme linéaire discret (ou entier ou mixte) (on notera PLNE, ou MIP).

$$\text{Maximiser } c_1^T x_1 + c_2^T x_2$$
$$\text{sous les contraintes}$$
$$A_1 x_1 + A_2 x_2 \leq b$$
$$x_1 \in \mathbb{R}^{n_1}$$
$$x_2 \in \mathbb{N}^{n_2}$$

où c_1, c_2 sont des vecteurs, A_1 et A_2 des matrices. x_1 la partie continue et x_2 la partie entière.

Programme linéaire

- Un problème linéaire continu peut être résolu en temps polynomial (*Khachiyan 1979*)
- Il existe des algorithmes polynomiaux efficaces pour résoudre un programme linéaire (ex : points intérieurs,*Karmarkar 1984*)
- L'algorithme du simplexe (*Dantzig 1947*) est le plus célèbre (et le plus efficace dans le cas général)
- Le simplexe n'est pas un algorithme polynomial !

- La PLNE est un problème NP-difficile
- Il est facile de montrer que la PLNE est un problème NP-difficile car de nombreux problèmes NP-difficiles peuvent être exprimés comme des PLNE.

Programme linéaire

- Solveurs de PL :
  - commerciaux Cplex (IBM), Xpress, Gurobi (microsoft), et même Matlab ou Excel...
  - académiques Lp de COIN-OR, Soplex de la ZIB ; et des solveurs libres comme Glpk (gnu).
  - Les meilleurs d'entre eux peuvent résoudre des PL jusqu'a 200.000 variables et 200.000 contraintes en quelques secondes.
- Solveurs de PLNE
  - les solveurs entiers sont beaucoup moins performants
  - Glpk par exemple ne dépassent pas quelques centaines de variables et contraintes
  - Cplex ou Gurobi sont les plus performants, quelques milliers milliers de variables / contraintes
  - cbc de coin-or est un des plus performants des solveurs open source (bien interfacé avec Python/Pyomo et Julia/JuMP)

Convexification

- Convéxification : rendre convexe ou améliorer la convexité
- Exemple :

$$q(x_1, x_2) = Cx_1 x_2 \ \forall (x_1, x_2) \in \{0, 1\}$$

avec C constante positive.
Non convexe. Mais :

$$\tilde{q}(x_1, x_2) = \frac{1}{2} C(x_1 + x_2)^2 - \frac{1}{2} C(x_1 + x_2) \ \forall (x_1, x_2) \in \{0, 1\}$$

est convexe (la partie quadratique de $\tilde{q}$ est ≥ 0. Et $q = \tilde{q}$.

Linéarisation

- Linérisation : méthodes permettant de transformer un PMD en un PL ou un PLNE
- Généralement, ces transformations demandent l'ajout de nombreuses variables supplémentaires.

- Nous allons voir quelques exemples dans les slides suivants.

Linéarisation

Exemple : une variable à valeurs dans un espace discret
Soit x une variable prenant sa valeur parmi les n possibilités $v_1, \ldots, v_n \in \mathbb{R}$.
Trouver une linéarisation de x.

Linéarisation

Exemple : une variable à valeurs dans un espace discret
Soit x une variable prenant sa valeur parmi les n possibilités $v_1, \ldots, v_n \in \mathbb{R}$.
Trouver une linéarisation de x.
On pose n variables de décisions binaires $y_1, \ldots, y_n$ tel que $y_i = 1$ si $x = v_i$. et 0 sinon.
On modélise alors par les deux contraintes :

$$x = \sum_{i=1}^{n} v_i y_i \text{ et } \sum_{i=1}^{n} y_i = 1$$

Linéarisation

L'opérateur "ou" numérique

On a une variable $x = 0$ ou $x \geq L$.
Avec L et x bornés par une valeur M.

Linéarisation

L'opérateur “ou” numérique

On a une variable $x = 0$ ou $x \geq L$.

Avec L et x bornés par une valeur M.

On ajoute une variable $y \in \{0, 1\}$ et on ajoute :

$$x \geq Ly \text{ et } x \leq My$$

Linéarisation

Carré d'une variable binaire
Soit $x \in \{0,1\}$. Linéarier x^2.

Linéarisation

Carré d'une variable binaire
Soit $x \in \{0, 1\}$. Linéarier x^2.
La variable x est équivalente a la variable x^2.

Linéarisation

Produit de deux variables binaires

Soit $x \in \{0, 1\}$ et $y \in \{0, 1\}$, on veut obtenir une variable $e = xy$.

Linéarisation

Produit de deux variables binaires
Soit $x \in \{0,1\}$ et $y \in \{0,1\}$, on veut obtenir une variable $e = xy$.
On a le résultat suivant :

$$e = xy \Leftrightarrow \begin{cases} e \leq x \\ e \leq y \\ e \geq x + y - 1 \\ e \geq 0 \\ e \in \mathbb{R} \end{cases}$$

Linéarisation

Généralisation :

- produit d'une variable binaire par une variable entière (bornée)
- produit de plusieurs variables binaires
- au carré de variables binaires,...

Toute forme quadratique peut se ramener a un PLNE binaire ! Mais en ajoutant de nombreuses variables et contraintes.

Modélisation par PLNE

Slides suivants : exemples de modélisation d'évènements a, b et c avec variables de décision binaires x_a, x_b et $x_c \in \{0; 1\}$:

Modélisation par PLNE

a et b ne peuvent pas se produire tous les deux :

Modélisation par PLNE

a et b ne peuvent pas se produire tous les deux : $x_a + x_b \leq 1$

Modélisation par PLNE

au moins un événement parmi a et b doit se produire :

Modélisation par PLNE

au moins un événement parmi a et b doit se produire : $x_a + x_b \geq 1$

Modélisation par PLNE

si a se produit alors b doit se produire :

Modélisation par PLNE

si a se produit alors b doit se produire : $x_b \geq x_a$

Modélisation par PLNE

si b ne se produit pas, alors a ne doit pas se produire :

Modélisation par PLNE

si b ne se produit pas, alors a ne doit pas se produire : $x_b \geq x_a$

Modélisation par PLNE

si a se produit, alors b et/ou c doivent se produire :

Modélisation par PLNE

si *a* se produit, alors *b* et/ou *c* doivent se produire : $x_b + x_c \geq x_a$

Modélisation par PLNE

si a ne se produit pas, alors une quantité y doit être nulle, sinon y est libre dans $\mathbb{R}$ (on suppose qu'on connait M tel que $y \leq M$:

Modélisation par PLNE

si a ne se produit pas, alors une quantité y doit être nulle, sinon y est libre dans $\mathbb{R}$ (on suppose qu'on connait M tel que $y \leq M$:

- On fixe M telle que $y \leq M$ à l'optimum du problème
- Un tel M existe, car sinon le problème est non borné
- modéliser par : $y \leq Mx_a$

Modélisation par PLNE

Maximiser la valeur minimale d'un ensemble de fonctions lineaires

- on veut maximiser la plus petite valeur prise par un ensemble de fonctions linéaires $a_i x, i = 1, \ldots, m$

Modélisation par PLNE

Maximiser la valeur minimale d'un ensemble de fonctions lineaires

- on veut maximiser la plus petite valeur prise par un ensemble de fonctions linéaires $a_i x, i = 1, \ldots, m$
- ajouter une variable z
- ajouter les contraintes $z \leq a_i x, i = 1, \ldots, m$
- Maximiser z

Modélisation par PLNE

k **contraintes parmi** n

- on a n contraintes $a_i x \leq b^i, i = 1, \ldots, n$
- k au moins doivent être satisfaites
- Pour chacune des contraintes on suppoe qu'il existe M_i tel que $a_i x \leq b_i + M_i$, soit satisfaite $\forall x$

Modélisation par PLNE

k contraintes parmi n

- on a n contraintes $a_i x \leq b^i, i = 1, \ldots, n$
- k au moins doivent être satisfaites
- Pour chacune des contraintes on suppoe qu'il existe M_i tel que $a_i x \leq b_i + M_i$, soit satisfaite $\forall x$
- On pose $y_1, \ldots, y_n$ des variables binaires

$$a_i x \leq b_i + M_i y_i \ \forall i = 1, \ldots, n$$

$$\sum_{i=1}^{n} y_i = n - k$$

Modélisation par PLNE

Implication entre contraintes

- Soit $a_1x \leq b_1$ et $a_2x \leq b_2$ deux contraintes
- Si $a_1x \leq b_1$ n'est pas satisfaite, alors $a_2x \leq b_2$ doit être satisfaite
- On suppose qu'il existe M tel que $-a_1x \leq -b_1 + M$ et $a_2x \leq b_2 + M$ soient vérifiées $\forall x$

Modélisation par PLNE

Implication entre contraintes

- Soit $a_1x \leq b_1$ et $a_2x \leq b_2$ deux contraintes
- Si $a_1x \leq b_1$ n'est pas satisfaite, alors $a_2x \leq b_2$ doit être satisfaite
- On suppose qu'il existe M tel que $-a_1x \leq -b_1 + M$ et $a_2x \leq b_2 + M$ soient vérifiées $\forall x$
- on ajoute une variable de décision binaire y
- on ajoute :

$$-a_1x \leq -b_1 + M(1-y)$$
$$a_2x \leq b_2 + My$$

Si $a_1x > b^1$, alors $y = 0$, et donc...