Programmation linéaire en nombres entiers

Formulation

Hugues Talbot

Laboratoire A2SI

20 mars 2009

Plan

Nombres entiers vs. nombres réels

- Jusqu'à présent, nous avons vu des problèmes de PL avec contraintes et des variables en nombres réels (positifs pour le simplexe)
- Une manière d'étendre le problème est d'exiger qu'un ou l'autre de ces aspects du problème s'exprime en nombres entiers.
- Dans le cas où les contraintes *et* les variables sont toutes deux entières, on parle de programmation linéaire en nombre entiers (ou simplement programmation en nombres entiers - IP en anglais).
- Dans le cas ou seulement un de ces aspects s'exprime en nombre entiers, ou même seulement certaines des variables, on parle de programmation linéaire mixte (MP en anglais).

Exemple

- Maximiser $z = x_1 + x_2$
- avec

$$\begin{array}{rcrcr} -2x_1 & + & 2x_2 & \geq & 1 \\ -8x_1 & + & 10x_2 & \leq & 13 \end{array}$$

- et $x_1, x_2 \geq 0$
- Avec le simplexe, en nombres réels, on trouve l'optimum $\{x_1 = 4, x_2 = 9/2\}$.
- Si le problème est contraint en nombres entiers (x_i sont des unités indivisibles), alors l'optimum est $\{x_1 = 1, x_2 = 2\}$!
- Comment passer de la solution continue à la solution entière ? Ici prendre l'entier le plus proche n'est pas faisable...

Catégories de problèmes en IP

1. Problèmes avec entrées/sorties discrètes : production d'objets, etc.
2. Problèmes avec conditions logiques : ajout de variables entières avec des contraintes supplémentaires. (par exemple : si le produit A est fabriqué alors produire également B ou C)...
3. Problèmes combinatoires : séquençage, allocation de ressources, emplois du temps, TSP : formulables en IP.
4. Problèmes non linéaires : souvent formulables en IP. C'est utile en particulier quand la région réalisables est non-convexe.
5. Problèmes de réseaux, problèmes de graphes – exemple : colorier une carte.

Remarques sur la formulation en IP

- La plupart des modèles sont dans la seconde catégorie (pb avec conditions logiques), la plupart des problèmes sont donc des problèmes mixtes avec quelques variables entières ajoutées ;
- IP peut être un moyen (avec de l'expérience) pour modéliser effectivement une grande classe de problèmes : contrairement à ce qu'on pourrait penser, contraindre davantage le problème (IP vs. LP) permet de modéliser plus de problèmes !
- Cependant, modéliser n'est pas résoudre ! On verra que IP est en général NP-complet, et donc les méthodes de résolutions sont non-linéaire en complexité...
- La complexité de la résolution dépend de la formulation !

Variables de décision

- Parmi les variables entières, il existe la classe des variables booléennes, pouvant prendre comme valeur 0 ou 1.
- Ces variables sont souvent utilisées en MP pour représenter des décisions : implications, liens entre variables, etc.
- Exemple des variables d'affectation et indicatrices.

Exemple : Choix

- Soient 4 choix possibles, chacun nécessitant des moyens et donnant un rendement

choix	moyen	rendement
1	5 unités	16 unités
2	7 ...	22 ...
3	4 ...	12 ...
4	3 ...	8 ...

- Moyens disponibles : 14 unités
- Maximiser le rendement.
- Extentions :
  1. On doit effectuer au plus 2 choix
  2. Choix 2 ne peut être effectué que si 1 est également pris

Affectation : formulation

- On utilise des variables booléennes (0 et 1), par exemple

$$X_j, j \in \{1, \ldots, 4\} = \begin{cases} 1 & \text{Choix effectué} \\ 0 & \text{Sinon} \end{cases}$$

- On cherche à maximiser $z = 16x_1 + 22x_2 + 12x_3 + 8x_4$
- Avec la contrainte

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

- Contraintes optionnelles :
  1. $x_1 + x_2 + x_3 + x_4 \leq 2$
  2. $x_1 \geq x_2 \Leftrightarrow -x_1 + x_2 \leq 0$

Variables indicatrices

- On suppose que x modèle une quantité (réelle) d'un ingrédient à inclure dans un mélange. On cherche à distinguer le cas $x = 0$ du cas $x > 0$.
- On introduit la variable δ, qui prend la valeur 1 lorsque $x > 0$, avec la contrainte :

$$x - M\delta \leq 0,$$

  avec M un coefficient connu représentant une valeur maximale possible de x.
- Avec cette contrainte, on a bien $x > 0 \Rightarrow \delta = 1$.

Indicatrices : suite

- Pour l'implication opposée ($x = 0 \Rightarrow \delta = 0$), c'est plus problématique. Cette implication est équivalente à $\delta = 1 \Rightarrow x > 0$.
- On introduit une implication moins sévère : $\delta = 1 \Rightarrow x > m$, avec m un niveau minimum acceptable en dessous duquel on considère x non-utilisé (dépendant de l'application). Une contrainte équivalente est alors :

$$x - m\delta \geq 0.$$

Problèmes à charge fixe

- Exemple : unités de production d'énergie
- Cas général de production avec coût initial et coût marginal.
- La puissance générée vaut soit $P_i = 0$, soit $P_i^m \leq P_i \leq P_i^M$.
- avec un coût :

$$C_i = \begin{cases} 0 & \text{Unité arrêtée} \\ a_i + b_i P_i & \text{Sinon} \end{cases}$$

- Produire une certaine puissance au moindre coût.
- Ici le coût est non-linéaire et même non-continu. LP n'est pas capable de modéliser le problème.
- Comment faire ?

Formulation de la charge fixe en IP

- On introduit la variable

$$x_i = \begin{cases} 0 & \text{Unité } i \text{ arrêtée} \\ 1 & \text{Sinon} \end{cases}$$

- Les contraintes

$$x_i P_i^m \leq P_i \leq x_i P_i^M$$

- avec

$$C_i = a_i x_i + b_i P_i$$

  (et non $C_i = a_i x_i + b_i P_i x_i$, qui serait non-linéaire).
- Note : la condition $P_i = 0, x_i = 1$ est réalisable mais de coût supérieur au cas $P_i = 0, x_i = 0$.

Problèmes d'affectation

- On a n tâches affectée à n personnes ;
- On veut affecter une et une seule tâche à chaque personne ;
- Le rendement de l'affectation de la tâche i à la personne j est donnée par la matrice C_{ij} ;
- On veut maximiser le rendement ;
- Formulation ?
- Formulation étendue : si le nombre de tâches est inférieur au nombres de personnes.

Formulation de l'affectation

- On introduit les variables x_{ij}

$$x_{ij} = \begin{cases} 1 & \text{tâche i affectée à la personne j} \\ 0 & \text{Sinon} \end{cases}$$

- contraintes :

$$\begin{aligned} \textstyle\sum_{i=1}^{n} x_{ij} &= 1 (\text{une seule tâche affectée à } j) \\ \textstyle\sum_{j=1}^{n} x_{ij} &= 1 (\text{chaque tâche affectée une fois}) \end{aligned}$$

- Maximiser

$$\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

Problème d'affectation étendu

- Nombre de personnes $>$ nombre de tâches, i.e n personnes, m tâches, $n > m$: $\sum_{i=1}^{m} x_{ij} \leq 1, j = 1, \ldots, n$
- Somme des tâches affectées à la personne j est ≤ 1 : $\sum_{j=1}^{n} x_{ij}, i = 1, \ldots, m$
- Somme des personnes affectées à la tâche i est $= 1$: on ajoute des variables d'écart : $s_j + \sum_{i=1}^{m} x_{ij} = 1, j = 1, \ldots, n$
- Maximiser

$$\sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

Autre façon de formuler : pb de transport

Nous verrons comment reformuler ce problème dans le dernier cours.

- (diagramme personne-tâches 4.18).
- contraintes en entiers,
- valeurs max de x_{ij} est 1, min = 0, donc solution en nombre entiers.

Conversion algèbre booléenne en algèbre linéaire avec variables binaires

- on introduit

$$\delta_i = \begin{cases} 1 & \text{si } x_i \text{ est vrai} \\ 0 & \text{sinon} \end{cases}$$

-

$$\begin{array}{rcl} x_i \quad \text{vrai} & \equiv & \delta_i = 1 \\ x_i \quad \text{faux} & \equiv & \delta_i = 0 \end{array}$$

Opérations logiques

$\sim$	négation
$\wedge$	ET logique
$\vee$	OU logique
$\Rightarrow$	Implication
$\Leftrightarrow$	Equivalence
$\oplus$	OU exclusif
V	VRAI
F	FAUX

Quelques opérations

$$
\begin{array}{lcr}
x_1 \vee x_2 = V & \equiv & \delta_1 + \delta_2 \geq 1 \\
x_1 \wedge x_2 = F & \equiv & \delta_1 = 1, \delta_2 = 1 \\
\sim x_1 = 1 & \equiv & \delta_1 = 0 \\
x_1 \Rightarrow x_2 & \equiv & \delta_2 \geq \delta_1 \\
x_1 \Leftrightarrow x_2 & \equiv & \delta_1 = \delta_2 \\
x_1 \oplus x_2 = V & \equiv & \delta_1 + \delta_2 = 1
\end{array}
$$

Combinaison de variables logiques et continues

- Variable logique = indicateur (e.g. ouvert/fermé , chaud/froid)
- cas $x_i = [f(x_i) \leq 0]$, soit, si $\delta_i = 1$ si $x_i = V$.
-
$$\begin{array}{lll} \delta_i = 1 & \text{si} & f(x_i) \leq 0 \\ \delta_i = 0 & \text{si} & f(x_i) \geq \varepsilon \text{ (précision)} \end{array}$$
- On considère
$$\begin{array}{lll} M & = & \max_{x_i}(f(x_i)) \\ m & = & \min_{x_i}(f(x_i)) \end{array},$$

  $M > 0, m < 0.$

Equations algébriques

$$
\begin{aligned}
f(x_i) &\leq M(1-\delta_i) \\
f(x_i) &\geq \varepsilon + (m-\varepsilon)\delta_i
\end{aligned}
$$

1. soit $\delta_i = 1$, alors $f(x_i) \leq 0, f(x_i) \geq \varepsilon$.
2. soit $\delta_i = 0$, alors $f(x_i) \leq M, f(x_i) \geq \varepsilon$.
3. soit $f(x_i) \geq 0$, alors $\delta_i = 0$ (démo tableau)
4. soit $f(x_i) \leq 0$, alors $\delta_i = 1$ (idem)

Produit de variables binaires

- soit le produit de variables $\delta_3 = \delta_1 \delta_2$: contraintes non-linéaires.
- Pour transformer en série de contraintes linéaires, on pose :

$$\begin{aligned} \delta_3 &\leq \delta_1, & \delta_3 &\leq \delta_2 \\ \delta_3 &\geq \delta_1 + \delta_2 - 1 \end{aligned}$$

On vérifie l'équivalence :

1. $\delta_1 = 0, \delta_2 = 0 \Rightarrow \delta_3 \leq 0 \Rightarrow \delta_3 = 0$
2. $\delta_1 = 1, \delta_2 = 0 \Rightarrow \delta_3 \leq 0, \delta_3 \geq 0 \Rightarrow \delta_3 = 0$
3. de même pour $\delta_1 = 0, \delta_2 = 1$
4. $\delta_1 = 1, \delta_2 = 1 \Rightarrow \delta_3 \leq 1, \delta_3 \geq 1 \Rightarrow \delta_3 = 1$
5. Inversement : $\delta_3 = 1 \Rightarrow \delta_1 \geq 1, \delta_2 \geq 1, \delta_1 + \delta_2 \leq 2$, donc $\delta_1 = \delta_2 = 1$.
6. $\delta_3 = 0 \Rightarrow \delta_1 \geq= 0, \delta_2 \geq 0, \delta_1 + \delta_2 \leq 1$ donc seulement δ_1 ou δ_2 vaut 1 (l'autre vaut 0).

Produit d'une variable binaire et d'une variable continue

- soit le produit non-linéaire de deux variables, l'une continue, l'autre booléenne : $\delta f(x)$, i.e :

$$y = \begin{cases} 0 & \text{si } \delta = 0 \\ f(x) & \text{si } \delta = 1 \end{cases}$$

- On impose les contraintes :

$$\begin{cases} y & \leq M\delta \\ y & \geq m\delta \\ y & \leq f(x) - m(1-\delta) \\ y & \geq f(x) - M(1-\delta) \end{cases}$$

Produit booléen-continu, suite

1. soit $\delta = 0$, alors $y \leq 0, y \geq 0$, donc $y = 0$.
   Egalement $f(x) - M \leq y \leq f(x) - m$, donc $f(x) - M \leq 0$ et $f(x) - m \geq 0$, ce qui est non-critique.
2. soit $\delta = 1$, alors

$$\left.\begin{array}{ll} y & \leq M \text{ (non-critique)} \\ y & \geq m \text{ (non-critique)} \\ y & \leq f(x) \\ y & \geq f(x) \end{array}\right\} y = f(x)$$

3. L'inverse est également correct : $y = f(x) \Leftrightarrow \delta = 1$.

Applications

- Systèmes dynamiques et logiques
  Exemple :

$$X(t+1) = \begin{cases} 0.8X(t) & +U(t) & \text{si } X(t) \geq 0 \\ -0.8X(t) & +U(t) & \text{si } X(t) < 0 \end{cases}$$

  avec $-10 \leq X \leq 10$, $-1 \leq U \leq 1$.
- Systèmes invariants linéaires par morceaux
- Machines à états finis.

Quelques problèmes en optimisation combinatoire

1. P.L. en nombres entiers :

$$\begin{array}{ll} \min & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \quad \leq \mathbf{b} \\ & x_i \text{ entiers} \end{array}$$

2. Cas particulier : variables booléennes

$$\begin{array}{ll} \min & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \quad \leq \mathbf{b} \\ & x_i = \ 0 \text{ ou } 1, \mathbf{x} \in \mathcal{B} \end{array}$$

Problèmes-types - I

3. Sac à dos (knapsack) : une seule contrainte

$$\begin{array}{lll} \max & \mathbf{C}^T\mathbf{x} & \\ & \mathbf{a}^T\mathbf{x} & \leq b \\ & \mathbf{x} \in \mathcal{B} & \end{array}$$

Note : $\mathbf{a} > 0$, sinon, $x_i' = 1 - x_i$.

4. Sacs à dos multiples

$$\begin{array}{lll} \max & \mathbf{C}^T\mathbf{x} & \\ & \mathbf{AX} & \leq \mathbf{b} \\ & \mathbf{x} \in \mathcal{B} & \end{array}$$

Note : $\forall i, j, A_{ij} \geq 0$.

Problèmes-types - II

5. Couplage (set matching)

$$\begin{array}{ll} \max & \mathbf{C}^T\mathbf{x} \\ & \mathbf{AX} \quad \leq 1(\text{ vecteurs de 1s }) \\ & \mathbf{X} \in \mathcal{B} \end{array}$$

Note : $\forall i, j, A_{ij} \in \mathcal{B}, X \in \mathcal{B}$.

6. Recouvrement (set covering)

$$\begin{array}{ll} \min & \mathbf{C}^T\mathbf{x} \\ & \mathbf{AX} \quad \geq 1(\text{ vecteurs de 1s }) \\ & \mathbf{X} \in \mathcal{B} \end{array}$$

Note : $\forall i, j, A_{ij} \in \mathcal{B}, X \in \mathcal{B}$.

Problèmes-types - III

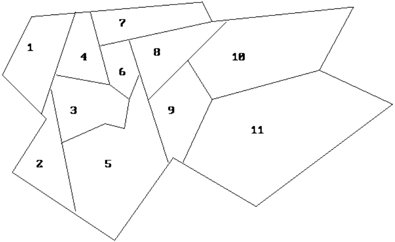
7. Partitionnement (set partitionning)

$$\begin{aligned} \min \quad & \mathbf{C}^T\mathbf{x} \\ & \mathbf{AX} \quad = 1(\text{ vecteurs de 1s }) \\ & \mathbf{X} \in \mathcal{B} \end{aligned}$$

Note : $\forall i, j, A_{ij} \in \mathcal{B}, X \in \mathcal{B}$.
Cas particulier : problème d'affectation.

Exemple de problème de recouvrement

Soit la ville suivante, composée des arrondissements suivants :



On doit construire un certain nombre d'hôpitaux, qui devront s'occuper des urgences médicales provenant de l'arrondissement où il est construit et de tous ses voisins immédiats. On cherche à minimiser le nombre hôpitaux. Formulation ?

Formulation du problème des hôpitaux

- On se donne une variable x_i booléenne pour chaque arrondissement. Si x_i vaut 1, il y aura un hôpital dans l'arrondissement i, et si elle vaut 0, il n'y en aura pas.
- On formule de la façon suivante :
  min $\sum_i x_i$, avec :

$$
\begin{array}{lcl}
x_1 + x_2 + x_3 + x_4 & \geq & 1 \\
x_1 + x_2 + x_3 + x_5 & \geq & 1 \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & \geq & 1 \\
x_1 + x_3 + x_4 + x_6 + x_7 & \geq & 1 \\
x_2 + x_3 + x_5 + x_6 + x_8 + x_9 & \geq & 1 \\
x_3 + x_4 + x_5 + x_6 + x_7 + x_8 & \geq & 1 \\
x_4 + x_6 + x_7 + x_8 & \geq & 1 \\
x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} & \geq & 1 \\
x_5 + x_8 + x_9 + x_{10} + x_{11} & \geq & 1 \\
x_9 + x_{10} + x_{11} & \geq & 1
\end{array}
$$

- On demande de "recouvrir" l'ensemble de manière optimale : une solution est $x_3 = x_8 = x_9 = 1$ et le reste à 0.

Conclusion

- On peut modéliser de nombreux problèmes par la programmation linéaire en nombres entiers (IP) ou par la programmation linéaire mixte (MP).
- Problèmes logiques, d'affectation, réseau, etc.
- PI et PM sont donc des outils puissants de modélisation
- Reste à résoudre ces problèmes :
  - existence d'une solution ?
  - unicité ?
  - algorithme de calcul de la solution ?
- On verra que la plupart des problèmes de PI et PM sont des problèmes difficiles à résoudre (NP-difficile).
- Deux méthodes principales de résolution : par coupes (Gomorry) et par séparation-évaluation (Branch-and-bound).