Sujet 5: Programmation stochastique — résolution des problèmes avec recours

MSE3313: Optimisation Stochastiqe

Andrew J. Miller

Dernière mise au jour: October 26, 2011

Dans ce sujet...

1. Algorithme dit "L-shaped"
   - Reformulation de Benders
   - Définition de l'algorithme
   - Comparaison avec les bornes déjà définis
2. Programmation stochastique pour les problèmes de multiples étapes: un exemple
3. Rémarques sur la réduction de variance

Rappel

Le problème de recours avec une distribution finie:

$$\begin{array}{ll} \min & c^T x + \mathcal{Q}(x) \\ \text{s.à.} & Ax = b, x \in X \end{array}$$

où

$$\mathcal{Q}(x) = \sum_j p^j Q(x, \xi^j),$$

l'ensemble X est défini par les bornes et/ou par des spécifications que quelques-un des variables x soient entières, et

$$Q(x, \xi) = \begin{array}{ll} \min & q(\xi)^T y \\ \text{s.à.} & W(\xi)y = h(\xi) - T(\xi)x, y \in Y \end{array}$$

Ici encore, l'ensemble Y est défini par les bornes simples sur une variables et/ou par des spécifications que quelques-un des variables y soient entières.

Formulation déterministe équivalent

$$
\begin{array}{ll}
\min & c^T x + \sum_j p^j (q(\xi^j)^T y^j) \\
\text{s.à.} & Ax = b \\
& T(\xi^j)x + W(\xi^j)y^j = h(\xi^j), \forall j \\
& x \in X;\ y^j \in Y, \forall j
\end{array}
$$

On appelle ce modèle **FDE**.
On va utiliser la structure *bloque-diagonale* de la matrice des contraintes pour résoudre ce problème.

1. Algorithme dit “L-shaped”
   - Reformulation de Benders
   - Définition de l’algorithme
   - Comparaison avec les bornes déjà définis
2. Programmation stochastique pour les problèmes de multiples étapes: un exemple
3. Rémarques sur la réduction de variance

Convexité dans les variables de première étape

On a besoin de la première supposition du dernier sujet:

Supposition (1)

$Y = \{y : y \geq 0\}$

Avec cette supposition, on a vu que

$$
\begin{aligned}
Q(x, \xi^j) &= \left\{ \begin{array}{ll} \min_y & q(\xi^j)^T y \\ \text{s.à.} & W(\xi^j) y = h(\xi^j) - T(\xi^j) x \\ & y \geq 0 \end{array} \right. \\
&= \left\{ \begin{array}{ll} \max_\pi & (h(\xi^j) - T(\xi^j) x)^T \pi \\ \text{s.à.} & W(\xi)^T \pi \leq q(\xi) \\ & \pi \text{ libre} \end{array} \right. ,
\end{aligned}
$$

pour chaque scènario j.

Reformulation du dual de $Q(x, \xi^j)$

De la théorie de la programmation linéaire, on sait que, s'il existe une solution optimale, alors il existe un **point extrème** qui est une solution optimale.

Soit

$$\mathcal{P}(j) = \{\pi^{j1}, ..., \pi^{jN(j)}\},$$

l'ensemble de tous les points extrèmes du dual de $Q(x, \xi^j)$.
On peut reformuler alors

$$\begin{aligned} Q(x, \xi^j) &= \left\{ \begin{array}{ll} \max_\pi & (h(\xi^j) - T(\xi^j)x)^T \pi \\ \text{s.à.} & W(\xi)^T \pi \leq q(\xi) \\ & \pi \text{ libre} \end{array} \right. \\ &= \max_{\pi^{jn} \in \mathcal{P}(j)} (\{h(\xi^j) - T(\xi^j)x)^T \pi^{jn}\} \\ &= \max_{n \in \{1,...,N(j)\}} \{(h(\xi^j) - T(\xi^j)x)^T \pi^{jn}\}. \end{aligned}$$

C'est à dire qu'on peut énumerer tous les points extrèmes possible, et on peut choisir celui qui maximise la fonctionne objective.

Reformulation du dual (suite)

Pas une très bonne idée pour un algorithme...mais est-ce qu'on peut utiliser ces idées pour generer seulement les point extrèmes nécessaires pour resoudre le problème?

Théoreme

$$\begin{aligned} Q(x,\xi^j) &= \max_{n\in\{1,\ldots,N(j)\}} \{h(\xi^j) - T(\xi^j)x)^T \pi^{jn}\} \\ &= \begin{cases} \min & q_j \\ s.à. & q_j \geq h(\xi^j) - T(\xi^j)x)^T \pi^{jn}, n = 1, \ldots, N(j) \end{cases} \end{aligned}$$

Une variable. $N(j)$ contraintes.

Exemple : Installation de capacité

A considerer : l'ensemble des points extrèmes duals pour chaque scénario.

Generation dynamique des points extrèmes du dual

Il est practiquement impossible de générer tous les points extrèmes des duals des problèmes de recours.

On va alors générer dyamiquement les points extrèmes dont on a besoin.

L'algorithme (I)

Algorithme

Pas 1 *Trouver une solution de première étape initiale* x^1*.*
Soit $K = 1$*.*

Pas 2 *Evaluer* $Q(x^K, \xi^j)$ *pour chaque scénario* j*.*
$z_P^K := c^T x^K + \sum_j p_j Q(x^K, \xi^j)$, **une borne supérieure**.

Pas 3 *Ajouter la contrainte*

$$q_j \geq (h(\xi^j) + T(\xi^j)x)^T \pi^{jK}$$

au problème maître, pour chaque j*.*

A remarquer: dans Pas 3 de la Kème iteration, π^{jK} est la solution optimale du dual trouvé dans l'évaluation de $Q(x^K, \xi^j)$.

L'algorithme (II)

Algorithme

Pas 3 *Ajouter la contrainte*

$$q_j \geq (h(\xi^j) + T(\xi^j)x)^T \pi^{jK}$$

au problème maître, pour chaque scenario j.

Pas 4 *Resoudre le problème maître:*

$$\begin{array}{ll} \min & c^T x + \sum_j p^j q_j \\ s.à. & Ax = b, x \in X \\ & q_j \geq (h(\xi^j) + T(\xi^j)x)^T \pi^{jk}, \forall j, k = 1, ..., K \end{array}$$

Soit $\{x^{K+1}, q^{K+1}\}$ *la solution optimale. Alors* $z_D^K := c^T x^{K+1} + \sum_j p^j q_j^{K+1}$ *est une***une borne inférieure***.*

Pas 5 *Si le valeur de* z_P^K *est suffisament proche à* z_D^K*, terminer. Sinon, retourner à Pas 2.*

Finitude de l'algorithme

Impliqué par le Théorème de convexité de chaque problème de recours $Q(x, \xi^j), j = 1, ..., J$, et par le fait que le nombre des points extrêmes d'un polyèdre est toujours fini.

Exemples

- Installation des capacities
- Planification des chirurgies
- Localisation des dépôts
- Planification de production (exercise maison)

1. Algorithme dit "L-shaped"
   - Reformulation de Benders
   - Définition de l'algorithme
   - Comparaison avec les bornes déjà définis
2. Programmation stochastique pour les problèmes de multiples étapes: un exemple
3. Rémarques sur la réduction de variance

Supérieures

Solution réalisable

Edmundson-Mandansky

Inférieures

Jensen

Scénarios indépendants

1. Algorithme dit "L-shaped"
   - Reformulation de Benders
   - Définition de l'algorithme
   - Comparaison avec les bornes déjà définis

2. Programmation stochastique pour les problèmes de multiples étapes: un exemple

3. Rémarques sur la réduction de variance

1. Algorithme dit “L-shaped”
   - Reformulation de Benders
   - Définition de l’algorithme
   - Comparaison avec les bornes déjà définis
2. Programmation stochastique pour les problèmes de multiples étapes: un exemple
3. Rémarques sur la réduction de variance

Réduction de nombre de scénarios

A souvenir

- La décomposition "L-shaped": pourquoi cela définit un algorithme exact et fini
- Les bornes générées par l'algorithme "L-shaped": pourquoi ils sont valides
- L'exemple de problèmes de multiples étapes: planification de production