Description d'article

Armand Fouquiau, Romany Stéphane

Université Paris-Sud

Octobre 2017

But de l'article

Cette article a pour but de de combler le vide littéraire concernant une famille de programmes stochastique : Stochastic Mixed-Integer Convexe Programs.

Toutes les décisions prises pour chaques scénarios à l'instant $t-1$ sont prises en compte à l'instant t. La fonction d'association $X(\xi_s) = x_s$ implique que x_s dépend des variables aléatoires de la distribution ξ.

La prise de décision se fait suivant des scénarios des sous problèmes de la forme : $f(x_s,\ \xi_s) = min\ \{f_s(x_s) \mid x_s \in C_s,\ x_s \in \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z}\}$
Avec $n = n_r + n_z$, la taille du vecteur x_s

L'algorithme Progressive Hedging (Haies progressive) a été inventer en 1991 par R. Rockafellar et West, L'algorithme Progressive Hedging (Haies progressive) a été inventer en 1991 par R. Rockafellar et West,

Généralités

Si on relaxe l'ensemble des contraintes de nonaticipativité noté ici $\mathcal{N}$, On peut convertir le problème P en problème convexe.

Accordingly, letting $\mathcal{Y}$
represent the set of feasible dual multipliers, the ordinary Lagrangian, achieved through the dualization of the constraint $X - \hat{X} = 0$.

On introduit des multiplicateurs Lagrangiens $\lambda_s \, \forall s = 1 \ldots S$
En multipliant ces multiplicateurs par les probabilités p_s pour chaque scénario s, on obtient $p_s\lambda_s$. Ces valeurs doivent être interpretées comme les multiplicateurs duals des contraintes de nonanticipativité accocié au scénario s.
La somme de tous les coefficients $p_s \cdot \lambda_s = 0$

On interprète l'équation (1) comme une extention du Lagrangien $\mathcal{L}(X, \hat{X}, \Lambda)$.
Avec les termes :
$\|x_s - \hat{x}_s\|_2^2$ qui permet une meilleur prise de décision,
$\left\|\lambda_s - \lambda_s^{k-1}\right\|_2^2$ avec k, un itération de l'algorithme PH,
ρ, une constante positive qui accumule l'impacte des deux précédentes quantités, au fil des itérations de l'algorithme PH.

L'algorithme PH règle certains problèmes de séparabilités due à la présence des termes $\hat{x}_s$

Les optimisations peuvent être faites pour chaque scénario indépendant, au regard de chaque variable $(X, \hat{X}, \Lambda)$

Progressive hedging

On instaure $\hat{x}^0$ en tant que minimum de la fonction objective $f(x_s,\ \Xi_s)$.
Cela leur permet d'obtenir une borne inférieure.

Ici mettre le code de l'algo.

L'algorithme de base PH heuristique peut trouver des valeurs qui ne respectent pas les contraintes de nonanticipativité, alors que l'algorithme PH-BAB écarte les solutions ne respectant pas ses dernières.
Pour chaque nouveaux noeuds crées, σ^{+}, σ^{-} on garde la même contrainte de réalisabilité : $\mathrm{Q}^{\sigma} = \mathrm{Q}^{\sigma^{+}} = \mathrm{Q}^{\sigma^{-}}$

Les meilleurs solvers du marché peuvent détecter la consistance de C_s^σ c'est à dire calculer en temps fini $\mathcal{N} \cap (\times_{s=1}^{S} C_s^\sigma)$ pour déterminer la réalisabilité de chaque relaxations de la fonction objective $z(\sigma)$
Avec $C_s^\sigma \Leftrightarrow C_s \cap Q_s^\sigma$: l'ensemble des x_s réalisables pour chaque scénario individuel.

Branch-and-Bounds

Cas des programmes stochastiques à deux niveaux

La partie intéressante est de considérer que les programmes stochastiques Mixed-integer convexes ne contiennent que des variables binaires au premier niveau. C'est à dire que les vecteurs x_s du premier niveaux sont de la forme $x_1 = (0, 1, 1, 0 \dots)$.

$\mathcal{N}$ n'est pas significatif pour le deuxieme niveau. On peut construire les conditions de nonanticipativité en se basant sur les solutions du premier niveau, pour chaque scénario seul.

En général, quand on décompose un scénario, l'algorithme PH-BAB peut s'arréter sur des noeuds où les conditions de nonanticipativité $\mathcal{N}$ sont garanties.

Calcul des performances

Pour tester leurs algorithmes, l'équipe utilise les ressources de serveurs dit SSLP (benchmarks pour solvers) contenant les données nécessaires pour l'exécution des programmes. Elle se base sur des problèmes linéaire et quadratique.
On utilise la notation SSLP n.m.S qui signifie qu'on propose un problemes linéaire à n variables binaires au premier et deuxième niveau et $n \times m$ variables binaires au deuxieme niveau. S est le nombre de scénario.
La notation SSLPQ n.m.S indique que les problèmes sont quadratiques.

Test sur quatre algorithme :

- PH heuristique (La version de base de Rockafellar and West seule).
- PH-BAB (Décrit dans l'algorithme 5).
- PH-BAB-Apx (Un version approximative de la précédente).
- CPLEX (Un solveur du constructeur IBM).

Conclusion