



UNIVERSITÉ DE MONTRÉAL

DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

Rapport Devoir 2

Nguyen-Xuan-Bach Bui

IFT6512 - Programmation stochastique

All codes are available at https://github.com/BNXBach/Work_stuffs_A2025/tree/main/TP_IFT

Partie 1

On veut prouver que le programme master dans la décomposition régularisée et la méthode de la région de confiance sont équivalents. On commence par écrire les 2 programmes en les notant programme A et B :

— Programme A :

$$\begin{aligned} \min \quad & c^T x + \sum_{k=1}^K \theta_k + \frac{1}{2} \|x - a^\nu\|_2^2 \\ \text{s.t.} \quad & Ax = b \\ & D_l x \geq d_l \\ & E_{l(k)} x + \theta_k \geq e_{l(k)} \\ & x \geq 0 \end{aligned}$$

— Programme B :

$$\begin{aligned} \min \quad & c^T x + \sum_{k=1}^K \theta_k \\ \text{s.t.} \quad & Ax = b \\ & D_l x \geq d_l \\ & E_{l(k)} x + \theta_k \geq e_{l(k)} \\ & \|x - a^\nu\|_2 \leq \Delta_\nu \\ & x \geq 0 \end{aligned}$$

Pour utiliser l'indice de l'exercice, on peut vite vérifier que :

- Les 2 fonctions objectives sont convexes (mentionnées dans [Fra11]).
- La contrainte d'égalité est affine.
- Les contraintes des coupes de réalisabilité et des coupes d'optimalité sont affines donc convexes.
- La contrainte de la région de confiance est convexe.

Alors sous la condition de Slater, les conditions de KKT sont suffisantes et nécessaires pour qu'une solution soit optimale pour ces 2 programmes.

Maintenant en reprenant les notations de l'indice (fonction f, g, h , etc...). On commence par le programme A :

$$\begin{aligned}
KKT_A &= \begin{cases} 0 \in \partial f_A(x_A^*) + \sum_{i=1}^m \lambda_i^* \partial g_i(x_A^*) + \sum_{j=1}^p \mu_j^* \Delta h_j(x_A^*) + N_X(x_A^*) \\ g_i(x_A^*) \leq 0, \quad h_j(x_A^*) = 0, \quad \lambda_i^* \geq 0, \quad \lambda_i^* g_i(x_A^*) = 0 \end{cases} \\
&= \begin{cases} 0 \in c + (x_A^* - a^\nu) + \sum_{i=1}^m \lambda_i^* \partial g_i(x_A^*) + \sum_{j=1}^p \mu_j^* \Delta h_j(x_A^*) + N_X(x_A^*) \\ g_i(x_A^*) \leq 0, \quad h_j(x_A^*) = 0, \quad \lambda_i^* \geq 0, \quad \lambda_i^* g_i(x_A^*) = 0 \end{cases}
\end{aligned}$$

Pour le programme B, on note d'abord qu'il a des mêmes contraintes que le programme A sauf la contrainte sur la région de confiance. C'est une inégalité alors sous les notations de l'indice, on a une fonction g en plus et la fonction objective f_B est aussi différente. De plus, pour simplifier le problème, on note aussi que la norme 2 est non négative et c'est une contrainte de type \leq alors on peut réécrire comme $\frac{1}{2}||x - a^\nu||^2 - \frac{1}{2}\Delta_\nu^2 \leq 0$. On a donc :

$$\begin{aligned}
KKT_B &= \begin{cases} 0 \in \partial f_B(x_B^*) + \sum_{i=1}^{m+1} \lambda_i^* \partial g_i(x_B^*) + \sum_{j=1}^p \mu_j^* \Delta h_j(x_B^*) + N_X(x_B^*) \\ g_i(x_B^*) \leq 0, \quad h_j(x_B^*) = 0, \quad \lambda_i^* \geq 0, \quad \lambda_i^* g_i(x_B^*) = 0 \end{cases} \\
&= \begin{cases} 0 \in c + \lambda_{m+1}^* \partial \left(\frac{1}{2}||x_B^* - a^\nu||^2 - \frac{1}{2}\Delta_\nu^2 \right) + \sum_{i=1}^m \lambda_i^* \partial g_i(x_B^*) + \sum_{j=1}^p \mu_j^* \Delta h_j(x_B^*) + N_X(x_B^*) \\ g_i(x_B^*) \leq 0, \quad h_j(x_B^*) = 0, \quad \lambda_i^* \geq 0, \quad \lambda_i^* g_i(x_B^*) = 0 \end{cases} \\
&= \begin{cases} 0 \in c + \lambda_{m+1}^*(x_B^* - a^\nu) + \sum_{i=1}^m \lambda_i^* \partial g_i(x_B^*) + \sum_{j=1}^p \mu_j^* \Delta h_j(x_B^*) + N_X(x_B^*) \\ g_i(x_B^*) \leq 0, \quad h_j(x_B^*) = 0, \quad \lambda_i^* \geq 0, \quad \lambda_i^* g_i(x_B^*) = 0 \end{cases}
\end{aligned}$$

Pour démontrer l'équivalence, on va montrer qu'une solution optimale du programme A est aussi optimale pour le programme B avec une valeur Δ_ν . Puis, avec Δ_ν fixé, on veut montrer qu'une solution optimale du programme B est aussi optimale pour le programme A. Alors, en conclusion, comme une solution optimale pour ce problème est unique (montré dans [Fra11]), les 2 problèmes ont la même solution optimale et donc ils sont équivalents.

C'est important de noter ici que dans la direction $(A) \Rightarrow (B)$, on peut choisir Δ_ν car on n'a pas encore fixé le programme B. Par contre, dans la direction $(B) \Rightarrow (A)$ on doit utiliser la même valeur Δ pour fixer le problème B à celui dans la direction précédente.

Direction $(A) \Rightarrow (B)$

Soit $(x_A^*, \theta_1^A, \dots, \theta_K^A)$ la solution optimale du programme A. Alors, les conditions KKT_A sont valides. On vérifie si les conditions KKT_B sont respectées. Si on garde toutes les coefficients λ et μ de KKT_A et on considère $\lambda_{m+1}^* = 1$, on retrouve la première ligne de la condition. Pour la deuxième ligne, il ne faut qu'observer cette condition :

$$\begin{aligned} & \frac{1}{2}\lambda_{m+1}^*(\|x_A^* - a^\nu\|_2^2 - \Delta_\nu^2) = 0 \\ \Rightarrow & \|x_A^* - a^\nu\|_2^2 = \Delta_\nu^2 \\ \Rightarrow & \|x_A^* - a^\nu\|_2 = \Delta_\nu \end{aligned}$$

Alors, avec cette valeur de Δ_ν , les conditions de KKT_B sont satisfaites et donc la solution optimale du programme A est aussi optimale pour le programme B où $\Delta_\nu = \|x_A^* - a^\nu\|_2$.

Direction $(B) \Rightarrow (A)$

Soit $(x_A^*, \theta_1^A, \dots, \theta_K^A)$ la solution optimale du programme A et $(x_B^*, \theta_1^B, \dots, \theta_K^B)$ la solution optimale du programme B et on fixe $\Delta_\nu = \|x_A^* - a^\nu\|_2$. On note d'abord que l'ensemble des contraintes du programme A partient à l'ensemble des contraintes du programme B. Alors, toutes solutions de B sont réalisables dans le programme A (B est plus stricte). Alors, comme $(x_A^*, \theta_1^A, \dots, \theta_K^A)$ est optimale pour le programme A, on a :

$$c^T x_A^* + \sum_{k=1}^K \theta_k^A + \frac{1}{2} \|x_A^* - a^\nu\|_2^2 \leq c^T x_B^* + \sum_{k=1}^K \theta_k^B + \frac{1}{2} \|x_B^* - a^\nu\|_2^2 \quad (1)$$

Puis, comme $(x_B^*, \theta_1^B, \dots, \theta_K^B)$ est une solution du programme B, on a cette relation en raison de la contrainte de la région de confiance :

$$\begin{aligned} & \|x_B^* - a_\nu\| \leq \Delta_\nu \\ \Rightarrow & \|x_B^* - a_\nu\| \leq \|x_A^* - a_\nu\| \end{aligned}$$

On a donc :

$$c^T x_B^* + \sum_{k=1}^K \theta_k^B + \frac{1}{2} \|x_B^* - a^\nu\|_2^2 \leq c^T x_B^* + \sum_{k=1}^K \theta_k^B + \frac{1}{2} \|x_A^* - a^\nu\|_2^2 \quad (2)$$

Ensuite, on a $\|x_A^* - a^\nu\|_2 \leq \|x_A^* - a_\nu\|_2$ (évident), alors $(x_A^*, \theta_1^A, \dots, \theta_K^A)$ est aussi réalisable pour le

programme B. De plus, comme $(x_B^*, \theta_1^B, \dots, \theta_K^B)$ est optimale pour le programme B, on a alors :

$$\begin{aligned} c^T x_B^* + \sum_{k=1}^K \theta_k^B &\leq c^T x_A^* + \sum_{k=1}^K \theta_k^A \\ \Rightarrow c^T x_B^* + \sum_{k=1}^K \theta_k^B + \frac{1}{2} \|x_A^* - a^\nu\|_2^2 &\leq c^T x_A^* + \sum_{k=1}^K \theta_k^A + \frac{1}{2} \|x_A^* - a^\nu\|_2^2 \end{aligned} \quad (3)$$

En combinant (1), (2) et (3), on note une chaîne d'inégalités de type \leq où la première et la dernière terme est la même. Alors, toutes les termes sont en fait égaux.

Si on reprend (1), on peut voir que $\min f_A = f_A(x_A^*, \theta_1^A, \dots, \theta_K^A) = f_A(x_B^*, \theta_1^B, \dots, \theta_K^B)$. Donc, $(x_B^*, \theta_1^B, \dots, \theta_K^B)$ est une solution optimale de A. \square

NOTE : Pour la direction $(B) \Rightarrow (A)$, on n'a pas utilisé les conditions KKT pour la montrer parce que c'est beaucoup plus difficile de montrer que $\lambda_{m+1}^* = 1$. Dans la direction $(A) \Rightarrow (B)$, on n'a pas besoin de faire la démonstration parce qu'on reprend seulement $\lambda_1, \dots, \lambda_m$ de KKT_A . Alors, on peut choisir λ_{m+1} pour que la première condition de KKT_B soit satisfaite. Dans la direction $(B) \Rightarrow (A)$, on doit reprendre $\lambda_1, \dots, \lambda_{m+1}$ de KKT_B et donc pour retrouver KKT_A , il faut qu'on montre que $\lambda_{m+1} = 1$.

Partie 2

Implémentation

L'implémentation de l'algorithme L-shaped multi-cut est dans le notebook TP2-Multicut.ipynb. Pour l'implémentation de l'algorithme avec régularisation, on a choisi de l'implémenter avec l'approche de la région de confiance en utilisant la norme infinie. La raison principale est parce que avec l'approche de régularisation normale, le programme master n'est plus linéaire et donc c'est plus difficile pour les solveurs à résoudre le problème. En fait, GLPK n'accepte pas du tout des programmes non linéaires et il faut utiliser HiGHS. De plus, un changement de point de départ peut affecter HiGHS et le solveur peut retourner des erreurs même si la solution est techniquement réalisable.

Avec l'approche de la région de confiance, le programme master reste linéaire et les contraintes de plus en sont aussi. Alors, c'est plus efficace pour les solveurs à résoudre même s'il y a plus de contraintes et la partie code en général est beaucoup plus stable. L'implémentation de l'algorithme L-shaped multi-cut avec région de confiance est dans le notebook TP2-TrustRegion.ipynb.

NOTE : Le notebook TP2-L2Norm.ipynb est aussi inclus dans le projet. Ce notebook n'est pas notre choix de l'implémentation pour l'exercice. On a inclu ce notebook comme une source de référence et pour montrer qu'on a rencontré les problèmes mentionés.

Comparaison

On va comparer entre la méthode L-shaped multi-cut et la méthode L-shaped avec région de confiance. Pour la deuxième méthode, on va comparer aussi la performance en changeant le point de départ a_ν . On a 2 cas :

- Cas naïve : matrice de 1 partout.
- Cas moyen (AVG) : la matrice est la solution optimale du problème d'espérance. On sait que cette solution est proche de celle du programme stochastique alors c'est un bon choix.

Les critères de comparaison sont le nombre d'itérations (et donc le nombre de coupes) et aussi la performance numérique des algorithmes. Voici le tableau de comparaison :

Méthode	Nb d'itérations	Temps(ms)	Nb d'allocations	Mémoire(MiB)
Mutli-cut	8	247.376	3842809	161.92
Trust region naïve	10	612.327	9117298	388.04
Trust region AVG	5	274.62	4546037	193.26

TABLE 1 – Comparaison entre les différentes versions de l'algorithme L-shaped

On note d'abord que l'approche de la région de confiance est la meilleure en fonction de l'efficacité (nombre d'itérations) mais cela dépend beaucoup du point de départ. Avec un mauvais point de départ, la méthode de région de confiance est moins efficace à l'approche L-shaped multi-cut simple. Cela coïncide avec [Fra11]

Par contre, numériquement, c'est l'approche L-shaped multi-cut simple qui est la meilleure. La raison est parce que dans l'approche région de confiance, il faut qu'on calcule ρ et fait plusieurs comparaisons en plus. C'est pourquoi même si on a moins d'itérations, la mémoire utilisée est plus grande dans l'approche région de confiance.

En terme de temps d'exécution, c'est vrai que l'approche de région de confiance prend plus de temps chaque d'itération mais on sait que c'est plus efficace. Le problème d'ici est que notre problème est encore assez simple alors la différence entre le nombre d'itérations des 2 méthodes est petite. Dans les cas où le problème est plus complexe, c'est possible que l'approche région de confiance peut sauver du temps en réduisant significativement le nombre d'itérations. Mais finalement, il faut qu'on résoud le problème d'espérance d'abord pour optimiser l'approche région de confiance alors si on considère la solution du problème d'espérance comme une étape obligatoire, c'est plus difficile de justifier l'approche région de confiance comme la meilleure solution.

En conclusion, c'est clair que la méthode région de confiance est plus efficace en terme de nombre d'itérations mais en terme de temps d'exécution/mémoire, le choix n'est pas clair et le meilleur choix dépend beaucoup du problème initial. Dans le cas de notre problème, c'est l'approche multi-cut simple qui gagne.

Références + Utilisation de l'intelligence artificielle

Références

- [Fra11] John R. Birge et FRANÇOIS LOUVEAUX. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer New York, NY, 2011. ISBN : 978-1-4614-0237-4.
- [Bas] Fabien BASTIN. *Notebook L-Shaped Example*. URL : https://github.com/fbastin/optim/blob/master/SP/two_stages.ipynb.

Utilisation de l'intelligence artificielle

J'ai utilisé l'IA pour :

- Expliquer les conditions KKT + condition de Slater de manière plus intuitivement.
- Faire des sommaires dans ([Fra11]).
- Aide avec des syntaxes de Julia pour gérer des contraintes dans le cas L-shaped région de confiance parce que les contraintes change avec chaque itération.
- Aide avec des syntaxes de Latex pour écrire ce rapport.

Le contenu de ce rapport est écrit sans utilisation de l'IA. La partie code Julia + les algorithmes sont basés sur les diapos du cours, [Fra11] et [Bas]