Génération de colonnes en décomposition croisée

Thibault VIGNON *(Ecole polytechnique, Palaiseau, France)*

Ce travail de recherche a été effectué au cours de mon stage dans le cadre d'une collaboration entre le LIPN, le LIP6 et EDF R&D, au sein d'un projet PGMO-IROE.

Ces recherches s'inscrivent dans le cadre théorique de la résolution de programmes linéaires en nombres entiers par des algorithmes de type séparation et évaluation (*Branch & Bound*). L'enjeu principal des méthodes appartenant à cette famille est de parvenir à calculer rapidement une borne de bonne qualité à chaque nœud. Afin d'obtenir des bornes meilleures que celle de relaxation linéaire, on emploie souvent des techniques de décomposition comme la génération de colonnes qui s'appuient fortement sur un choix d'inégalités à convexifier. Ce stage s'est intéressé à une technique que nous appelons *décomposition croisée*. Celle-ci consiste à simultanément considérer lors de la même décomposition plusieurs sous-structures qui se superposent. La décomposition croisée basée sur la relaxation Lagrangienne est connue dans la littérature, sous le nom de *variable splitting* ou *décomposition Lagrangienne* [1]. En revanche, la décomposition croisée basée sur la génération de colonnes est moins classique et a été assez peu mise en pratique : c'est cette approche que nous explorons ici.

La décomposition croisée consiste à convexifier simultanément deux ensembles de contraintes qui opèrent de manière différente sur une matrice de variables de décision : d'une part sur ses lignes, d'autre part sur ses colonnes, d'où le croisement inhérent à cette décomposition. On obtient ainsi deux sous-problèmes plus simples correspondant à chaque catégorie de contraintes. Puisque ces contraintes partagent les mêmes variables, pour décomposer on doit dupliquer les variables de décision. Le problème maître de génération de colonnes coordonne les deux sous-problèmes au moyen de contraintes qui lient leurs copies respectives des mêmes variables [2].

La décomposition croisée en génération de colonnes comporte deux enjeux principaux. Le premier est d'améliorer la valeur de relaxation obtenue par la décomposition choisie, à travers le choix des contraintes convexifiées. Le second enjeu est le temps nécessaire pour parvenir à cette valeur de relaxation, qui dépend à la fois de la complexité des sous-problèmes résolus à chaque étape et du nombre d'itérations nécessaires pour la convergence de la procédure. En général, ces deux enjeux rentrent en conflit. D'une part, dualiser de nouvelles contraintes augmente la complexité des sous-problèmes et donc le temps de calcul. D'autre part, on peut interrompre la génération de colonnes avant convergence, mais il faut alors se satisfaire d'une borne légèrement inférieure à la valeur de relaxation [3].

L'objectif qui motive une grande partie du travail présenté consiste donc à améliorer la vitesse de la convergence sans nuire à la valeur de relaxation. Pour ce faire, on s'est particulièrement intéressé à la dynamique itérative de la génération de colonnes. On établit dans le cas de la décomposition croisée une borne dynamique valide pour le problème initial, qui permet d'interrompre précocement la génération de colonnes. Une autre préoccupation de toute génération de colonnes est sa stabilisation. On propose sur ce plan une analyse approfondie de deux techniques qui sont par nature spécifiques à la double décomposition que nous employons : l'emploi d'une inégalité plutôt qu'une égalité dans le variable splitting, et la répartition des coûts entre les sous-problèmes. Ces outils techniques sont effleurés dans la littérature, mais ce travail est à notre connaissance le premier qui combine les deux : on introduit une méthode nouvelle qui conjugue efficacement l'apport de ces techniques.

Est présentée une première application qui illustre ce paradigme de décomposition : celui du Capacitated Facility Location Problem. On introduit une version modifiée de ce problème qui se prête à la mise en place d'une décomposition croisée, en y ajoutant une exigence de fiabilité des solutions. Le cœur de ce problème consiste à apparier des clients avec des usines ; décomposer selon l'axe des clients d'une part et selon l'axe des usines de l'autre nous fournit une décomposition croisée.

On aborde ensuite l'application de la décomposition croisée à laquelle la majorité du travail appliqué a été consacrée : le Unit Commitment Problem. Dans ce cas, les deux axes sur lesquels on décompose sont l'horizon temporel discrétisé et les différentes unités de production d'électricité. A été implémenté sur ce problème un code de résolution employant la génération de colonnes en décomposition croisée, intégrée à une architecture complète de Branch & Price. On détaille cette implémentation et les résultats expérimentaux qui en sont issus. Sont confirmés expérimentalement les gains en valeur de relaxation permis par la décomposition par rapport à la relaxation linéaire, et par la décomposition croisée par rapport à la décomposition simple. Est mis en évidence l'impact bénéfique conséquent des techniques d'accélération de la convergence que nous avons développées. Enfin, l'intérêt de la décomposition croisée en Branch & Price est confirmé par la résolution d'instances d'une taille qui est inaccessible à une approche par décomposition simple.

Références

[1] M. Guignard and S. Kim, "Lagrangean decomposition for integer programming : theory and applications," *RAIRO-Operations Research*, vol. 21, no. 4, pp. 307–323, 1987.

[2] L. Létocart, A. Nagih, and N. T. Moungla, "Dantzig-wolfe and lagrangian decompositions in integer linear programming," *Int. J. Math. Oper. Res.*, vol. 4, no. 3, pp. 247–262, 2012.

[3] F. Vanderbeck and L. A. Wolsey, "Reformulation and decomposition of integer programs," in *50 Years of Integer Programming 1958-2008*, pp. 431–502, Springer, 2010.