Génération de colonnes : une approche par apprentissage

Markus Kruber[1], Axel Parmentier[2], Marco Lübbecke[1]

[1] RWTH Aachen
Lehrstuhl für Operations Research, Kackertstraße 7, 52072 Aachen, Deutschland
[2] Université Paris Est, CERMICS (ENPC)
6 et 8 avenue Blaise Pascal Cité Descartes - Champs sur Marne 77455 Marne la Vallée Cedex 2
axel.parmentier@enpc.fr {kruber,luebbecke}@or.rwth-aachen.de

Mots-clés : *Décomposition de Dantzig-Wolfe, génération de colonnes, apprentissage supervisé, programme linéaire en nombres entiers (PLNE).*

Étant donné un PLNE et une décomposition bloc-diagonale de sa matrice des coefficients, la décomposition de Dantzig-Wolfe (DW) permet de reformuler ce *problème initial* en un nouveau PLNE appelé *problème maître* admettant les mêmes solutions. Le problème maître ayant typiquement un nombre exponentiel de variables, il est résolu par génération de colonnes. Le *problème esclave* consiste alors à identifier les variables à considérer. Lorsque la décomposition bloc-diagonale correspond à la structure sous-jacente du problème initial, le problème maître admet une bien meilleure relaxation linéaire que le problème initial. Il est donc bien mieux résolu que le problème initial par les approches de type Branch-and-Bound. Ceci fait de la génération de colonnes une des techniques importantes pour le traitement de PLNE de grande taille.

Différents types de décompositions sont utilisées dans la littérature. Par exemple, les décompositions blocs-diagonales avec une bordure, c'est à dire des contraintes liantes entre les blocs, sont très utilisées pour les problèmes de tournées de véhicules, tandis que les décompositions en escalier permettent de bien traiter les problèmes de sac-à-dos temporels [2]. En réorganisant les lignes et les colonnes, chacune de ces structures peut être artificiellement exhibée dans tout PLNE. Néanmoins, si une décomposition qui ne correspond pas à la structure du problème initial est utilisée, l'approche par génération de colonnes donne de mauvaises performances. Comme c'est généralement « l'utilisateur » qui connaît la structure du problème et fournit une bonne décomposition, la décomposition de DW n'est pas utilisée dans les solveurs PLNE généraux.

Des solveurs PLNE basés sur la décomposition de DW ont cependant été proposés ces dernières années. Ces solveurs s'appuient sur un solveur PLNE pour résoudre les problèmes maîtres et esclaves. Dans cette présentation, nous utilisons `GCG` [3] comme extension pour la génération de colonnes de `SCIP` [1]. Ces solveurs procèdent en trois étapes. Tout d'abord, un *détecteur* cherche des décompositions de DW du problème initial. La structure du problème n'étant pas connue a priori, différents types de décompositions sont recherchés. Les décompositions obtenues sont ensuite évaluées. Enfin, selon le résultat des évaluations, le problème est résolu soit par génération de colonnes en utilisant la meilleure décomposition, soit en résolvant directement le PLNE initial. Comme une approche par génération de colonnes ne fonctionne bien que si une décomposition de DW adaptée au PLNE considéré est utilisée, la question du choix de la méthode de résolution est cruciale dans la performance de tels solveurs. Cette question est actuellement traitée par le biais de critères évaluant la performance des détecteurs. Nous proposons d'y répondre par une approche de type *apprentissage supervisé*.

Pour cela nous commençons par répondre à la question, *étant donné une décomposition $\mathcal{D}$, vaut-il mieux l'utiliser dans un solveur par génération de colonnes ou utiliser le solveur PLNE sur le problème initial $\mathcal{P}$?* Dans cette perspective, nous extrayons du couple $(\mathcal{P}, \mathcal{D})$ un vecteur de statistiques suffisantes $\phi(\mathcal{P}, \mathcal{D})$, que nous fournissons ensuite à un modèle de classification binaire $f : \mathbb{R}^d \to \{0, 1\}$ qui fournit la réponse désirée. Notre vecteur $\phi(\mathcal{P}, \mathcal{D})$

Solver	SCIP	GCG	AS	OPT
Non opt.	23	25	17	14
CPU time (h)	48.2	53.3	36.8	30.0

TAB. 1 – Performances des solveurs sur un ensemble de 65 instances structurées et non structurées.

contient plus d'une centaine de statistiques suffisantes, et permet donc de tirer partie de plus d'information que celle rendue accessible par les critères actuellement utilisés pour évaluer la qualité des décompositions. Nous combinons ensuite ces modèles de classifications binaires f dans des approches « un-contre-tous » pour choisir si une approche par génération de colonnes doit être utilisée, et si oui quelle décomposition utiliser. Les modèles de classification f sont appris à partir d'une base de données contenant les performances du solveur sur un ensemble d'instances et de décompositions.

La performance d'une telle approche par apprentissage supervisé repose sur trois éléments. Tout d'abord, le vecteur de statistiques suffisantes $\phi(\mathcal{P}, \mathcal{D})$ doit contenir l'information pertinente. Ensuite, pour qu'un modèle statistique donné présente de bonnes performances sur un couple $(\mathcal{P}, \mathcal{D})$ donné, il faut que des PLNE et des décompositions comparables apparaissent dans l'ensemble de formation. Enfin, les bons modèles de classification doivent être utilisés. Dans notre cas, les meilleures performances ont été obtenues pour des machines à support de vecteur (SVM). Nous avons combinés des SVM dans une approche « un-contre-tous » pour traiter le problème du choix de la meilleure décomposition.

Le tableau 1 fournit le nombre d'instances non résolues à l'optimum et le temps de calcul total obtenus par les différentes approches sur un ensemble test composé de 65 PLNE structurés et non-structurés. Pour le produire, nous avons appris notre SVM sur une base de donnée composée de 345 PLNE et des décompositions détectées sur ces différents PLNE. Les trois-quart des PLNE utilisés dans les ensemble de formation et de test ont une une structure spécifique : ce sont par exemple des problèmes de coloration de graphes, de couverture d'ensemble ou de tournées de véhicules. Le reste des instances provient de la de la bibliothèque MIPLIB 2010 [4]. Les premières colonnes du tableau 1 fournissent les performances de `SCIP` et de `GCG` lorsqu'on laisse ce dernier choisir la décomposition. La colonne AS fournit les résultats obtenus en utilisant notre modèle par apprentissage supervisé pour le choix du solveur et de la décomposition à utiliser. Enfin, la colonne OPT fournit les résultats qui seraient obtenus si le meilleur solveur et la meilleure décompositions étaient choisis à chaque étape. Les temps de calcul incluent le temps de détection des différentes décompositions. Ces résultats montrent donc la pertinence d'une approche de type apprentissage supervisé pour le choix d'une décomposition, et plus généralement fournissent un argument en faveur de l'utilisation de méthodes à base de génération de colonnes dans les solveurs PLNE généraux. Une base de donnée de plus grande taille est en cours de construction, et des résultats numériques sur un ensemble de données de test plus importants seront présentés lors de la conférence.

Références

[1] Tobias Achterberg. Scip : solving constraint integer programs. *Mathematical Programming Computation*, 1(1) :1–41, 2009.

[2] Alberto Caprara, Fabio Furini, and Enrico Malaguti. Uncommon dantzig-wolfe reformulation for the temporal knapsack problem. *INFORMS Journal on Computing*, 25(3) :560–571, 2013.

[3] Gerald Gamrath and Marco E Lübbecke. Experiments with a generic dantzig-wolfe decomposition for integer programs. In *International Symposium on Experimental Algorithms*, pages 239–252. Springer, 2010.

[4] Thorsten Koch, Tobias Achterberg, Erling Andersen, Oliver Bastert, Timo Berthold, Robert E. Bixby, Emilie Danna, Gerald Gamrath, Ambros M. Gleixner, Stefan Heinz, Andrea Lodi, Hans Mittelmann, Ted Ralphs, Domenico Salvagnin, Daniel E. Steffy, and Kati Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2) :103–163, 2011.