

Linéarisation de variables ensemblistes ordonnées dans LocalSolver

Sandrine Bouguen

LocalSolver

24 avenue Hoche, 75008 Paris

sbouguen@localsolver.com

Mots-clés : *optimisation combinatoire, linéarisation, programmation linéaire mixte.*

1 Contexte

LocalSolver est un solveur d'optimisation mathématique de type « model & run ». Il intègre à la fois des méthodes heuristiques et des méthodes exactes afin d'obtenir de bonnes solutions rapidement et d'assurer leur qualité. Sur un même problème, plusieurs méthodes se lancent en parallèle pour exploiter les points forts de chaque algorithme. Parmi les méthodes exactes présentes dans le solveur, un solveur MIP permet de traiter les problèmes linéaires mixtes.

LocalSolver permet non seulement de modéliser des problèmes d'optimisation avec les variables de décision classiques de la programmation en nombres entiers : variables booléennes, entières et flottantes, mais dispose aussi de variables de collection, les listes et les ensembles. Ces variables permettent de stocker de façon ordonnée ou non un ensemble de valeurs. Elles facilitent la modélisation et améliorent les performances du solveur pour de nombreux types de problèmes, tel que les problèmes de tournées de véhicules, d'ordonnancement ou de remplissage.

2 Variable de *list* et problème de tournées de véhicules

Une variable ensembliste, appelée *list* dans le formalisme LocalSolver, est définie sur un ensemble discret (les entiers de 0 à $N - 1$) et représente un sous-ensemble ordonné de cet ensemble. Ce type de variable facilite notamment la modélisation de problèmes de tournées de véhicules.

Dans un problème de tournées de véhicules, on dispose d'une flotte de camions pour livrer N clients. Les camions doivent partir d'un dépôt et y revenir à la fin de la route. L'objectif est de construire des routes qui minimisent le nombre de camions utilisés et la distance totale parcourue. Ce problème peut se décliner de nombreuses façons, notamment en rajoutant des contraintes sur la capacité des camions ou sur les horaires d'ouverture des clients. Pour modéliser cette classe de problèmes avec LocalSolver, on utilise des listes. Pour chaque camion, une liste va contenir la séquence ordonnée des clients livrés par ce camion. Comme chaque client doit être visité une et une seule fois, les listes doivent former une partition du domaine $[0, N - 1]$. Pour cela, on utilise l'opérateur `partition` présent dans LocalSolver.

Cette formulation est très adaptée à la recherche par voisinage et permet d'obtenir de très bons résultats sur l'ensemble des problèmes de routing. Elle permet également de passer à l'échelle et de traiter les problèmes de grande taille.

3 Linéarisation automatique de modèles utilisant des listes

Notre objectif est de combiner la programmation linéaire mixte à la recherche locale sur les modèles utilisant des listes. Actuellement, le solveur MIP ne peut pas traiter les modèles à base

de listes, car cette représentation n'est pas linéaire. C'est pourquoi, nous souhaitons automatiser la reformulation linéaire de ces modèles. Le solveur MIP pourra ainsi traiter la formulation linéaire du problème, tandis qu'en parallèle la recherche locale se lancera sur la formulation avec liste. Il est important que cette reformulation soit automatique afin que l'utilisateur puisse continuer d'écrire la formulation avec liste, qui est la plus facile et intuitive. Il faut donc que la détection des modèles linéarisables se fasse sans paramétrage de l'utilisateur. De plus, l'objectif est de pouvoir appliquer la linéarisation au spectre le plus large possible de modèles afin de correspondre aux besoins variés de nos clients.

De nombreuses formulations existent pour linéariser les problèmes de tournées de véhicules [1]. En première approche, nous avons choisi d'utiliser la formulation de Miller-Tucker-Zemlin (MTZ) [2]. Cette dernière permet notamment d'éliminer les sous-tours sans ajouter un nombre exponentiel de contraintes.

Afin d'étudier l'intérêt de linéariser automatiquement les modèles à base de listes, nous avons comparé les résultats obtenus avec la formulation MTZ et avec la formulation avec liste dans LocalSolver sur 468 instances académiques de CVRPTW (*Capacited Vehicle Routing Problem with Time Windows*). Le tableau 1 présente le nombre d'instances où une des formulations est strictement meilleure que l'autre.

	Borne primale	Borne duale
Formulation liste	415	326
Formulation MTZ	6	142

TAB. 1 – Nombre d'instances améliorées par chaque formulation.

Nous observons que bien que la formulation liste soit meilleure globalement, il y a 142 instances où la borne duale est améliorée par la formulation MTZ. De même, on observe quelques instances où une meilleure borne primale est trouvée avec la formulation MTZ. Automatiser la linéarisation des listes permettra donc d'améliorer de nombreuses instances grâce à l'utilisation du solveur MIP en parallèle de la recherche locale.

Références

- [1] Roberto Roberti and Paolo Toth. Models and algorithms for the asymmetric traveling salesman problem : an experimental comparison. *EURO Journal on Transportation and Logistics*, 1(1-2) :113–133, 2012.
- [2] Clair E Miller, Albert W Tucker, and Richard A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4) :326–329, 1960.