

LocalSolver 9.5 : nouveautés et améliorations des performances

Julien Darlay

LocalSolver, 36 avenue Hoche, Paris, France

jdarlay@localsolver.com

Mots-clés : *solveur, recherche locale*

1 Introduction

LocalSolver est un solveur d'optimisation mathématique de type *model & run* [1]. Son formalisme d'entrée lui permet d'accepter tout modèle utilisant les opérateurs mathématiques usuels (arithmétiques, logiques, relationnels, etc.) avec des variables continues, entières ou ensemblistes. La recherche de solutions est faite par une approche heuristique basée sur des techniques de recherche locale dont l'objectif est de trouver rapidement des solutions de qualité. Des techniques d'optimisation globale sont utilisées en complément de la recherche locale pour accélérer la recherche de solutions, pour calculer des bornes et arrêter la recherche lorsque la solution optimale est trouvée.

La version 9.5 de LocalSolver, prévue pour début 2020 se concentre sur l'amélioration des performances générales du solveur. Cette présentation fera le tour des nouveautés importantes apportées au solveur sur la recherche de solutions et le calcul de bornes.

2 Recherche de solutions

L'amélioration de la recherche locale se concentre sur certaines familles de problèmes : les problèmes d'ordonnancements, les problèmes de tournées de véhicules avec fenêtres de temps et les problèmes à variables mixtes tels que le *unit commitment* ou la planification de production. La principale nouveauté de LocalSolver 9.5 concerne l'exploitation automatique de structures spécifiques à ces problèmes lors de l'évaluation du voisinage.

Pour les problèmes d'ordonnement cela se traduit par la détection d'une structure de précédences et l'exploitation de techniques de propagation issues de la programmation par contraintes pour réparer des solutions modifiées localement. Dans le cas des problèmes mixtes, la structure du sous-problème continue (souvent les quantités à produire) est détectée et la résolution par des techniques de programmation linéaire permet de réparer une solution modifiée par la recherche locale. Enfin, pour les problèmes de tournées de véhicules l'exploitation des expressions arithmétiques définissant les fenêtres de temps permet d'avoir des voisinages plus facilement réalisables. Ces structures sont détectées automatiquement par LocalSolver, ne réclamant à l'utilisateur aucun effort ni information supplémentaire à propos de son modèle.

L'exposé présentera des benchmarks sur chaque famille de problèmes pour illustrer l'impact de tous ces travaux. Les résultats présentés montrent une amélioration notable des performances avec bien plus de solutions faisables trouvées et de bien meilleure qualité (par exemple, un écart moyen à l'optimum réduit de 70% à 5% sur certains problèmes d'ordonnement).

3 Calcul de bornes

Le calcul de bornes a été introduit dans la version 8.5 de LocalSolver. Dans le même esprit que la recherche de solutions, le calcul de bornes se focalise sur l'obtention rapide de

bornes qui seront améliorées itérativement. LocalSolver reformule automatiquement le modèle de l'utilisateur et résout une succession de relaxations convexes pour calculer des bornes.

Les améliorations du calcul de bornes par optimisation globale se traduisent d'abord par l'amélioration des solveurs globaux : ajout d'un algorithme de *branch & reduce*, utilisation d'un algorithme de points intérieurs non linéaires, meilleure gestion de certains opérateurs, ajout de nouvelles techniques de preprocessing et de nouvelles coupes issues de l'état de l'art de la programmation linéaire en nombres entiers [2].

Le calcul de bornes a été étendu aux variables ensemblistes telles que les *sets* et les *lists*. Les listes permettent de modéliser de manière compacte des problèmes de permutations comme le problème du voyageur de commerce. Ces problèmes sont en pratique difficilement attaquables par des techniques d'optimisation globale mais des bornes peuvent être calculées par des techniques ad-hoc. Pour les problèmes de permutation, LocalSolver détecte automatiquement la structure du problème et utilise la relaxation lagrangienne de Held-Karp [3] pour calculer une borne inférieure. Cette méthode permet d'obtenir des écarts à l'optimalité de moins de 10% en 60 secondes pour les instances symétriques de la TSPLib jusqu'à 2000 sommets.

Références

- [1] F. Gardi, T. Benoist, J. Darlay, B. Estellon R. Megel Mathematical Programming Solver Based on Local Search. Wiley, 2014.
- [2] A. Alper, G. Nemhauser M. Savelsbergh Conflict graphs in solving integer programming problems. European Journal of Operational Research, 121(2000) :40-55.
- [3] M. Held Richard M. Karp The Traveling-Salesman Problem and Minimum Spanning Trees. Operations Research, 18.6(1970) : 1138-1162