

Hybridation de Feasibility Pump avec la recherche locale de LocalSolver

Tiphaine Rougerie

LocalSolver
36 avenue Hoche, 75008 Paris
trougerie@localsolver.com

Mots-clés : *Heuristiques primales, Feasibility Pump, recherche locale.*

1 Contexte

LocalSolver est un solveur d'optimisation mathématique basé sur différentes techniques de recherche opérationnelle : méthodes d'inférence pour réduire le problème lors du prétraitement, recherche locale pour fournir rapidement de bonnes solutions faisables, solveur MIP pour résoudre des instances linéaires, et solveur non linéaire afin de produire des solutions et des bornes inférieures de bonne qualité. L'hybridation de ces méthodes est très prometteuse, notamment par la synchronisation des solutions et des bornes trouvées par chaque solveur.

Les heuristiques primales ont pour but de fournir des solutions primal-faisables dès les premiers instants de la résolution. L'heuristique Feasibility Pump, qui consiste à résoudre des relaxations d'un problème MIP, a été hybridée avec la recherche locale de LocalSolver en tant que méthode d'amélioration des solutions trouvées, mais aussi pour aider à résoudre les points infaisables trouvés au cours de la résolution.

2 Problèmes étudiés

Dans un premier temps, l'heuristique Feasibility Pump a été implémentée pour les problèmes mixtes entiers (MIP), et utilise un algorithme du simplexe pour la résolution des sous-modèles LP obtenus par relaxation. Elle peut aussi être utilisée sur tout problème linéarisable par LocalSolver. L'heuristique sera étendue à l'avenir aux problèmes non linéaires, grâce au solveur NLP présent dans LocalSolver. Nous considérons un problème MIP formulé de la manière suivante :

$$(\text{MIP}) \begin{cases} \min c^T x & (1a) \\ \mathcal{A}x \leq b & (1b) \\ x_j \in \{0, 1\} & \forall j \in \mathcal{B} & (1c) \\ x_j \in \llbracket l_j, u_j \rrbracket \subseteq \mathbb{Z} & \forall j \in \mathcal{I} & (1d) \\ x_j \in \llbracket l_j, u_j \rrbracket \subseteq \mathbb{R} & \forall j \in \mathcal{R} & (1e) \end{cases}$$

où \mathcal{B} , \mathcal{I} et \mathcal{R} sont respectivement les ensembles d'indices des variables booléennes, entières et réelles. Les bornes l_j et u_j peuvent éventuellement être infinies.

3 Feasibility Pump

L'heuristique Feasibility Pump a été mise au point en 2005 par Fischetti, Glover et Lodi [3] pour des problèmes mixtes booléens, puis étendue aux problèmes mixtes entiers par Bertacco, Fischetti et Lodi [1]. Elle permet d'obtenir des solutions rapidement, en résolvant des sous-modèles LP faciles qui guident la recherche.

Une itération de Feasibility Pump, ou « *pump cycle* », consiste en deux étapes :

1. **Projection** : pour un point entier \tilde{x} donné, i.e. qui respecte les contraintes (1c), (1d) et (1e), le sous-problème (2) linéaire relâché permet d'obtenir la solution faisable la plus proche selon une distance Δ à définir. Les contraintes d'intégrité étant relâchées, la solution obtenue sera faisable mais non entière.

$$(\mathcal{LP}) \begin{cases} \min \Delta(x, \tilde{x}) \\ \mathcal{A}x \leq b & (1b) \\ x_j \in [0, 1] & \forall j \in \mathcal{B} \\ x_j \in [l_j, u_j] & \forall j \in \mathcal{I} \cup \mathcal{R} \end{cases} \quad (2)$$

2. **Arrondi** : l'intégrité est retrouvée (au détriment de la faisabilité) en arrondissant les x_j entiers et booléens, afin d'obtenir un nouveau point entier \tilde{x} donné pour initialiser l'étape 1 de l'itération suivante.

Deux directions de recherche sont ainsi explorées, l'une faisable (solutions x^*), l'autre entière (points \tilde{x}), jusqu'à trouver l'intersection des deux \tilde{x}^* , qui est une solution du problème. Le pseudo-code détaillé de Feasibility Pump est donné par Berthold [2].

4 Hybridation avec la recherche locale de LocalSolver

La recherche locale de LocalSolver a deux utilités pour Feasibility Pump : arrondir rapidement des points infaisables et améliorer localement les solutions faisables fournies par l'heuristique primale. Le meilleur point infaisable courant, c'est-à-dire celui de distance Δ non nulle et minimale, est régulièrement fourni à la recherche locale au cours de la résolution. Ce critère de distance minimale permet de choisir le point qui semble être le plus proche de l'intégrité. Pour plusieurs instances, la recherche locale peut rapidement trouver une solution entière faisable grâce à ce point initial, alors qu'elle ne trouve pas de solution sans ce point.

5 Benchmarks

L'heuristique Feasibility Pump a été implémentée au cœur de LocalSolver en tant que *thread* indépendant. Les tests ont été effectués sur plusieurs types de problèmes : les instances faisables de la Miplib 2010, des instances de *set cover* et *set partition* de la OR Library, ainsi que des instances de *bin packing* de la Bin Packing Library. Pour chaque instance, un *thread* de recherche locale est utilisé en parallèle de celui de l'heuristique primale sur une durée d'une minute.

Les performances de LocalSolver ont été améliorées sur plusieurs bibliothèques de problèmes : plus d'instances obtiennent des solutions faisables, certaines sont faisables en un temps plus court, et la qualité des solutions est meilleure pour un certain nombre de problèmes.

Références

- [1] Bertacco, L., Fischetti, M. et Lodi, A. (2007). « A feasibility pump heuristic for general mixed-integer problems ». *Discrete Optimization* 4.1, pages 63-76.
- [2] Berthold, T. (2006). « Primal heuristics for mixed integer programs ». Mémoire de master. Technische Universität Berlin.
- [3] Fischetti, M., Glover, F. et Lodi, A. (2005). « The feasibility pump ». *Mathematical Programming* 104.1, pages 91-104.