

Vers une programmation par recherche locale : LocalSolver 1.1

Thierry Benoist¹, Bertrand Estellon², Frédéric Gardi¹, Romain Megel¹, Karim Nouioua²

¹ Bouygues e-lab, Paris, France
{tbenoist, fgardi, megel}@bouygues.com

² Laboratoire d'Informatique Fondamentale - CNRS UMR 6166,
Faculté des Sciences de Luminy - Université Aix-Marseille II, Marseille, France
{bertrand.estellon, karim.nouioua}@lif.univ-mrs.fr

Mots-clés : *optimisation combinatoire, recherche locale, solveur/logiciel.*

Nous nous sommes engagés en 2007 dans un projet de recherche visant au développement d'un solveur de "programmation par recherche locale", c'est-à-dire tel que l'utilisateur modélise son problème et le solveur le résout par recherche locale. Notre but à terme est d'obtenir un formalisme mathématique déclaratif et un solveur adaptés au paradigme de recherche locale permettant de résoudre de façon autonome des problèmes d'optimisation combinatoire de grande taille, en particulier ceux dont l'échelle met en difficulté les solveurs classiques à base de recherche arborescente (PLNE, PPC). Première concrétisation de nos travaux présentée au congrès ROADEF 2010, le logiciel LocalSolver 1.0 permet de traiter efficacement les problèmes du type *assignment*, *partitioning*, *packing*, ou *covering* exprimés dans un formalisme aux variables de décision booléennes. D'abord testé sur un large éventail de problèmes académiques et industriels, le logiciel est aujourd'hui exploité à TF1 pour la résolution hebdomadaire d'un problème de placement de spots publicitaires. Preuve de son adéquation aux besoins du chercheur opérationnel, le logiciel a été utilisé en dehors du Groupe Bouygues par des ingénieurs de la société Eurodecision pour traiter plusieurs problèmes, dont l'optimisation énergétique d'un réseau de transport public. Il a également été utilisé par des chercheurs du LIF Marseille pour la construction des emplois du temps de l'université (cours et examens) et pour la construction automatique de résumés de réunions (traitement automatique du langage). Pour ces problèmes, LocalSolver produit en quelques dizaines de secondes des solutions que les meilleurs solveurs de PLNE n'obtiennent qu'après plusieurs heures de calcul ; dans certains cas, ces derniers ne produisent même aucune solution après plusieurs jours de calcul.

TAB. 1 – Résultats obtenus sur 9 instances représentatives du *steel mill slab design* avec un temps d'exécution limité à 1 minute sur un ordinateur double coeur cadencé à 3 GHz. Les résultats correspondent à la moyenne obtenue sur 5 essais (arrondie à l'entier supérieur). La notation "x" signifie qu'aucune solution admissible n'a été trouvée dans le temps imparti. L'optimum est obtenu par PLNE (CPLEX) après reformulation comme problème de *set partitioning* avec énumération explicite de l'ensemble des colonnes (sous-ensembles d'objets pouvant appartenir à une boîte).

	2-0	3-0	4-0	5-0	6-0	7-0	8-0	9-0	10-0
<i>Optimum</i>	22	5	32	0	0	0	0	0	0
LocalSolver 1.0	46	52	36	4	8	2	0	0	0
LocalSolver 1.1	36	9	36	0	4	1	0	0	0
CPLEX 12.2	136	288	x	126	x	232	226	163	133
CPO 2.3	90	65	58	50	54	46	28	29	20
Comet CBLS 2.1	136	135	69	65	42	30	26	21	20

La première partie de notre exposé sera consacrée aux avancées réalisées dans la version 1.1 de LocalSolver, qui sera rendue disponible à l'occasion du congrès ROADEF 2011. Des opérateurs

arithmétiques (valeur absolue, carré, modulo, division entière) ont été ajoutés au formalisme, permettant de modéliser simplement des problèmes fortement non linéaires en variables booléennes. Par ailleurs, les performances du solveur ont été améliorées. Le pool de mouvements autonomes, dont le but est de modifier localement la solution tout en maintenant sa faisabilité, a été enrichi de façon à accroître la diversification et à améliorer la convergence de la recherche locale. Ont notamment été ajoutés des mouvements permettant de cibler des variables de décision dont la modification est susceptible d'améliorer le coût de la solution courante. Des métaheuristiques comme le recuit simulé ont été ajoutées en sus de la descente standard, de façon à assurer une meilleure diversification de la recherche pour des temps d'exécution plus longs. Enfin, un mode d'exécution parallèle est implémenté, permettant d'exploiter des architectures multicœurs. Des résultats seront présentés sur un ensemble de benchmarks académiques et industriels, permettant de comparer les performances de LocalSolver avec des solveurs reconnus pour leur performance (IBM ILOG CPLEX pour la PLNE et IBM ILOG CPO pour la PPC) ainsi que certains langages ou frameworks récents visant à faciliter l'implémentation d'heuristiques de recherche locale (Dynadec Comet CBLIS). Le Tableau 1 ci-dessus présente un échantillon des résultats obtenus pour le *steel mill slab design* (problème de la CSPLib tiré d'une problématique industrielle de *bin-packing* avec boîtes de taille variable et contraintes d'exclusion mutuelle entre objets).

Dans une seconde partie, nous présenterons les fonctionnalités majeures actuellement en cours de développement, diffusées dans les prochaines versions du logiciel. Le formalisme LSP de LocalSolver sera enrichi à la manière des modeleurs de PLNE (AMPL, OPL, Mosel, etc.) afin de faciliter et raccourcir l'écriture des modèles. Ainsi, l'utilisateur disposera d'un véritable modeleur en sus du solveur. Par exemple, voici écrit dans ce nouveau formalisme le modèle du célèbre problème de *bin-packing* :

```
// data
nbObjets = ...; nbBoites = ...;
tailleObjets[1..nbObjets] = ...; tailleBoites[1..nbBoites] = ...;

// model
x[1..nbObjets][1..nbBoites] <- bool();
for[i in 1..nbObjets]
  constraint boolsum[j in 1..nbBoites]( x[i][j] ) == 1;
for[j in 1..nbBoites] {
  rempliBoites[j] <- sum[i in 1..nbObjets]( tailleObjets[i] x[i][j] );
  constraint rempliBoites[j] <= tailleBoites[j];
  residuBoites[j] <- if( rempliBoites[j] > 0, sum( tailleBoites[j], -rempliBoites[j] ), 0 );
}
minimize sum[j in 1..nbBoites]( residuBoites[j] );
```

Nous évoquerons également les travaux en cours concernant l'intégration de concepts ensemblistes tant au niveau du solveur qu'au niveau du modeleur. Ces concepts ensemblistes permettront à l'utilisateur de modéliser des problèmes comme le *bin-packing* à l'aide d'objets et d'ensembles, le rapprochant ainsi du modèle métier mais aussi du modèle naturel pour une résolution par recherche locale.

LocalSolver 1.1 peut être téléchargé sur les sites web du Bouygues e-lab ou du LIF Marseille, et utilisé gratuitement. Les binaires (exécutables et librairie C++ ISO) sont fournis pour les systèmes Windows, Linux, Mac OS X sous architecture x86. Des *wrappers* pour les langages Java et C# sont également disponibles. Un tutoriel de démarrage rapide est fourni pour la prise en main du solveur.