

Recherche locale haute performance

Bertrand ESTELLON¹, Frédéric GARDI², et Karim NOUIOUA¹

¹ Laboratoire d'Informatique Fondamentale, Faculté des Sciences de Luminy, Marseille
{karim.nouioua,bertrand.estellon}@lif.univ-mrs.fr

² Bouygues SA / DGITN / e-lab, Paris
fgardi@bouygues.com

1 Introduction

La recherche locale est une technique très utilisée en pratique pour aborder des problèmes d'optimisation combinatoire difficiles. L'idée, naturelle, est d'améliorer de façon itérative une solution existante en explorant un voisinage de celle-ci. Les solutions dites voisines sont généralement obtenues en appliquant des transformations (aussi appelées mouvements) plus ou moins importantes (c'est-à-dire plus ou moins locales) à la solution courante.

Après un premier essor dans les années 70 lié au travaux de Lin et Kernighan [8] sur le problème du voyageur de commerce, la recherche locale a connu un regain d'intérêt à la fin des années 90 avec de nombreuses études expérimentales et quelques tentatives d'analyse théorique [1]. Ce regain d'intérêt, et la popularisation qui s'en est suivie dans les milieux académiques et industriels, est sans doute dû à l'avènement du concept de métaheuristique. En effet, le nombre de papiers discutant de métaheuristique se compte en milliers ; des pionnières aux plus récentes, nous en avons dénombré plus d'une quinzaine dans la littérature [6]. Ainsi, la plupart des cours donnés sur le sujet se contente d'un état-de-l'art des métaheuristiques et ne mentionne la notion de recherche locale qu'au travers de quelques applications (par exemple le mouvement 2-opt pour le problème du voyageur de commerce). Pourtant, lorsqu'elles ne reposent pas tout simplement sur le concept de recherche locale, toutes ces métaheuristiques ne sont réellement performantes qu'une fois couplées avec de la recherche locale (par exemple les algorithmes à base de fourmis et les algorithmes génétiques) [1,2,10].

Nous affirmons que le concept de métaheuristique est malheureusement devenu un leurre, dissimulant l'essentiel de ce qui fait de la recherche locale une méthode performante en pratique : les voisinages et l'algorithmique sous-jacente. Nous proposons une méthodologie guidant la conception et l'implémentation d'heuristiques à base de recherche locale visant la performance, voire la haute performance. Nous illustrerons nos propos par deux exemples concrets d'application : l'ordonnement de véhicules dans les usines RENAULT (Challenge ROADEF 2005) [3,4] et la planification des opérations de maintenance chez FRANCE TÉLÉCOM (Challenge ROADEF 2007) [5].

2 Algorithmique haute performance

Notre définition de la performance est la suivante. Nous considérons comme performant un algorithme capable de fournir en temps raisonnable (quelques minutes) une solution de qualité. La plupart des besoins en optimisation rencontrés dans l'industrie s'exprime en ces termes ; lorsque les instances considérées sont de très grandes tailles (par exemple un voyageur de commerce parcourant plus d'un million de villes), un temps d'exécution de plusieurs heures peut être envisagé. Un algorithme performant doit également répondre à certaines exigences en matière d'ingénierie logicielle (fiabilité, robustesse, portabilité, maintenabilité).

La notion de haute performance est souvent assimilée à l'implémentation d'algorithmes sur une architecture à haute performance (en particulier sur une architecture parallèle). En fait, un algorithme haute performance reprend simplement la définition que nous avons donné dans le paragraphe précédent, en y ajoutant des superlatifs : algorithme capable de fournir une solution de très grande qualité en un temps très court. De même, l'exigence de robustesse est redoublée : un tel algorithme doit être capable de traiter convenablement des instances dégénérées. Pour plus de détails sur ce sujet, nous invitons le lecteur à consulter l'article de Moret [9] et les travaux remarquables d'Helsgaun [7] sur le problème du voyageur de commerce.

3 Méthodologie

Nous préconisons d’aborder la conception d’une heuristique à base de recherche locale suivant un cadre en trois parties : 1) heuristique de recherche, 2) mouvements, 3) algorithmique & implémentation. Nous estimons que la performance d’une heuristique à base de recherche locale est fonction à *part égale* du bon traitement de chacun de ces trois points.

L’heuristique, généralement spécifique au problème, correspond à la stratégie générale de recherche. Sa conception passe par la définition d’une fonction objectif qui favorise la convergence de la recherche locale, en assurant notamment une certaine densité des solutions dans l’espace de recherche. C’est aussi à ce niveau que peuvent être intégrées des fragments de métaheuristiques (seuils, listes tabou, *etc*). Toutefois, nous préconisons l’utilisation d’une simple descente dans un premier temps. En effet, la diversification des solutions doit se faire en premier lieu par les mouvements, et non seulement par la stratégie de recherche, chose qui est rarement mentionnée dans la littérature. C’est pourquoi nous préconisons également l’emploi d’une stratégie de type “first-improvement” avec choix des mouvements stochastiques.

Les mouvements sont donc au cœur de l’heuristique. Leur définition doit permettre de visiter un maximum de solutions (différentes) dans le temps imparti. Nous insistons sur l’aspect diversification, qui nous paraît plus crucial que l’aspect intensification. À ce niveau, les mouvements génériques de type échanges, insertions, *etc* ne suffisent pas : un grand nombre de mouvements, plus ou moins spécialisés, doivent être définis. L’utilisation de propriétés structurelles, spécifiques au problème (voir par exemple le travail d’Helsgaun [7] sur le voyageur de commerce ou [3] pour le problème d’ordonnancement de véhicules), permet de définir des mouvements ayant une probabilité de succès plus élevée.

Enfin, l’algorithmique, en particulier liée à l’évaluation des mouvements, est déterminante quant à l’effectivité de la recherche locale. La recherche locale est une technique de recherche incomplète, par opposition aux techniques de recherche complète (arborescente) ; néanmoins, son effectivité n’en demeure pas moins liée au volume de solutions visitées. Ainsi, une algorithmique incrémentale, basée sur l’exploitation d’invariances dans les structures de données, peut diviser le temps de convergence par 10. De même, une implémentation travaillée, soucieuse du principe de localité présidant la gestion de la mémoire cache et optimisée par *profiling* du code, peut encore accélérer par 10 le temps de convergence [9]. En définitive, il n’est pas surprenant de constater un facteur 100, voire 1000, entre les temps de convergence de deux heuristiques relevant pourtant des mêmes principes.

Références

1. E. Aarts, J.K. Lenstra (1997). *Local Search in Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Chichester, England, UK.
2. M. Dorigo, L.M. Gambardella (1997). Ant Colony System : a cooperative learning approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation* 1(1), pp. 53–66.
3. B. Estellon, F. Gardi, K. Nouioua (2005). Two local search approaches for solving real-life car sequencing problems. (à paraître dans *European Journal of Operational Research*)
4. B. Estellon, F. Gardi, K. Nouioua (2006). Large neighborhood improvements for solving car sequencing problems. *RAIRO Operations Research* 40(4), pp. 355–379.
5. B. Estellon, F. Gardi, K. Nouioua (2007). Une heuristique à base de recherche locale pour la planification de tâches avec affectation de ressources. In *ROADEF 2007*. Grenoble, France.
6. F.W. Glover, G.A. Kochenberger (2002). *Handbook of Metaheuristics*. International Series in Operations Research and Management Science, Vol. 57, Kluwer Academic Publishers, Norwell, MA.
7. K. Helsgaun (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research* 126(1), pp. 106–130.
8. S. Lin, B.W. Kernighan (1973). An effective heuristic algorithm for the Traveling-Salesman Problem. *Operations Research* 21, pp. 498–516.
9. B.M.E. Moret (2002). Towards a discipline of experimental algorithmics. In *Data Structures, Near Neighbor Searches, and Methodology : 5th and 6th DIMACS Implementation Challenges* (sous la direction de M.H. Goldwasser, D.S. Johnson et C.C. McGeoch), DIMACS Monographs, Vol. 59, pp. 197–213. American Mathematical Society, Providence, RI.

10. C. Solnon, V.-D. Cung, A. Nguyen, C. Artigues (2005). The car sequencing problem : overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. (à paraître dans *European Journal of Operational Research*)