

Méthodes Hybrides et Solutions Partielles dans Scatter Search

Igor Crévits¹, Semya Elaoud², Saïd Hanafi¹ et Christophe Wilbaut¹

¹ LAMIH, Université de Valenciennes, F 59313 Valenciennes Cedex9, France
{igor.crevits,said.hanafi,christophe.wilbaut}@univ-valenciennes.fr

² GIAD – FSEG – Sfax, Tunisie
samyaelaoud@yahoo.fr

Mots-Clés : *Sac à dos, Scatter Search, Relaxation, Solutions Partielles*

La recherche dispersée (scatter search (**SS**) selon la terminologie anglaise) est une méthode évolutive qui repose sur la gestion d'un ensemble (une population) référence qui doit permettre de conserver la diversité et la qualité des solutions rencontrées. La forme originale date de la fin des années 70 [2], et de nombreuses applications plus récentes peuvent être recensées [1, 4]. Laguna et Martí [6] ont décrit dans leur livre quelques principes intéressants pour une implémentation de SS. Glover a donné dans [3] un modèle en 5 composants majeurs pour SS et sa généralisation, *path relinking*. Dans ce travail, nous montrons qu'il est possible d'améliorer un algorithme SS standard en utilisant des solutions partielles dans différentes phases de l'approche. Nous proposons ainsi de nouveaux générateurs de solutions basés à la fois sur les solutions partielles et des composants d'une Heuristique Itérative basée sur la Relaxation (**HIR**) [8]. Nous validons cette approche sur le problème du sac-à-dos multidimensionnel (**MKP**) qui peut être formulé comme suit :

$$(\text{MKP}) \max \left\{ \sum_{j=1}^n c_j x_j : \sum_{j=1}^n a_{ij} x_j \leq b_i, \forall i \in M = \{1, 2, \dots, m\}, x_j \in \{0, 1\}, \forall j \in N = \{1, 2, \dots, n\} \right\}$$

avec n le nombre d'objets et m le nombre de contraintes. Toutes les données c, b, a sont supposées entières et non négatives. Le MKP est souvent utilisé comme benchmark pour tester des méthodes générales d'optimisation combinatoires. Le lecteur peut se référer à [5] pour plus de précisions sur le sujet. A notre connaissance, les meilleures solutions connues sur un ensemble d'instances corrélées et difficiles¹ sont dues à des approches hybrides [7, 8].

Plutôt que de manipuler uniquement des solutions binaires, nous envisageons dans ce travail d'intégrer des solutions partielles dans SS. Ainsi, au lieu de fixer toutes les variables à 0 ou 1 durant la génération des solutions diverses, nous proposons deux générateurs qui construisent une population de solutions partielles composées de 0, 1 et "*". Cette relaxation des contraintes d'intégralité permet de garder un contrôle supplémentaire lors du processus de génération. Une autre motivation dans l'utilisation des solutions partielles réside dans l'idée, bien connue en optimisation, qu'une réduction du problème jusqu'à un certain point permet de résoudre à l'optimum des problèmes NP-Difficiles en des temps raisonnables. Le premier générateur de solutions partielles repose sur un processus déterministe tiré d'un générateur existant pour les problèmes en variables 0-1 [3]. Le second repose sur l'utilisation d'une relaxation pour définir itérativement les variables qui seront libres dans les solutions construites.

¹<http://people.brunel.ac.uk/~mastjbj/jeb/info.html>

Lorsque la population de solutions partielles est générée, nous pouvons alors appliquer un opérateur de projection qui va générer une ou plusieurs solutions dans $\{0, 1\}^n$ à partir de chaque solution, en résolvant de façon exacte ou approchée chaque problème réduit associé aux variables libres de la solution. Cette phase de construction peut ainsi être résumée comme suit :

1. $Pop := \emptyset$; Construire Pop_{part} une population de solutions partielles ;
2. **pour** chaque solution $x \in Pop_{part}$ **faire**
 - (a) Construire P_x comprenant une ou plusieurs solutions à partir de x ;
 - (b) $Pop := Pop \cup P_x$;

Parmi les autres composants de notre algorithme, la phase d'amélioration repose sur une recherche locale simple qui cherche à remplacer un objet par un ou plusieurs autres dans la solution courante. La génération des sous-ensembles de solutions repose sur la méthode présentée dans [3]. Nous utilisons plusieurs phases de combinaison de sous-ensembles de solutions selon leur taille : une méthode simple de type path relinking pour ceux de taille 2, et une méthode basée sur un score pour les autres [6]. Finalement, nous intégrons un processus d'intensification basée sur les solutions partielles lorsque l'ensemble référence converge, de manière à exploiter l'information contenue dans cet ensemble avant de relancer la recherche.

Nous avons validé notre approche sur les 270 instances de la OR-Library référencées précédemment. Les résultats, qui seront donnés en détail lors de la conférence, montrent que notre algorithme converge rapidement vers des solutions de bonne qualité et proches des meilleures solutions connues. En particulier, nos heuristiques obtiennent 212 meilleures bornes inférieures sur les 270 instances en moins de 5 ou 10 minutes selon la taille des instances.

Ces résultats encourageants semblent montrer qu'il est possible de bien améliorer un algorithme de type SS en intégrant l'utilisation de solutions partielles. De plus, d'autres éléments, telle que l'utilisation de la mémoire dans la recherche, laissent entrevoir des perspectives d'amélioration.

Références

- [1] P. Belfiore et H. T. Yoshida Yoshizaki. Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. *European Journal of Operational Research*, 199:750–758, 2009.
- [2] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8:156–166, 1977.
- [3] F. Glover. A Template for Scatter Search and Path Relinking. In J.K. Hao, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers, editors, *Artificial Evolution, Lecture Notes in Computer Science*, 1363, pp. 13–54. Springer, 1998.
- [4] L. M. Hvattum et F. Glover. Finding local optima of high-dimensional functions using direct search methods. *European Journal of Operational Research*, 195:31–45, 2009.
- [5] H. Kellerer, U. Pferschy et D. Pisinger. *Knapsack Problems*. Springer, 2004.
- [6] M. Laguna et R. Martí. *Scatter Search. Methodology and Implementations in C*. Kluwer Academic Publishers, 2003.
- [7] M. Vasquez et Y. Vimont. Improved results on the 0–1 multidimensional knapsack problem. *European Journal of Operational Research*, 165:70–81, 2005.
- [8] C. Wilbaut et S. Hanafi. New convergent heuristics for 0–1 mixed integer programming. *European Journal of Operational Research*, 195:62–74, 2009.