

# Recherche Tabou à base de décomposition arborescente pour le problème de bin packing avec conflits

Ali Khanafer, François Clautiaux, El-Ghazali Talbi

INRIA Lille Nord-Europe, Université des Sciences et Technologies de Lille  
40 avenue Halley, 59650 Villeneuve d'Ascq, France

Ali.Khanafer@inria.fr, Francois.Clautiaux@univ-lille1.fr El-ghazali.Talbi@lifl.fr

**Mots-Clés :** *bin packing avec conflits, décomposition arborescente, recherche tabou.*

## 1 Introduction

Plusieurs problèmes de la vie réelle peuvent être modélisés sous la forme d'un *problème de bin packing avec conflits* (BPC), introduit par Jansen *et al.* [1]. Dans ce problème, on dispose d'une liste d'articles et d'un graphe de compatibilité, et on cherche à minimiser le nombre de conteneurs nécessaires pour placer un ensemble d'articles de telle manière que chacun des bins ne contient que des objets compatibles entre eux.

Nous proposons une approche heuristique pour résoudre le problème de BPC en deux dimensions. Cette approche est basée sur le concept de *décomposition arborescente* (voir figure 1), que nous appliquons au graphe de compatibilité de notre problème. Cela nous permet de décomposer l'instance en différents *clusters* qui sont ensuite résolus indépendamment par des heuristiques classiques. Dans ce travail, on utilise l'heuristique qu'on appelle BLC (Bottom Left with Conflicts), une adaptation de l'heuristique BL proposé par Coffman *et al.* [4] afin de prendre en compte les conflits entre les objets.

Dans une décomposition arborescente, un sommet peut appartenir à plusieurs clusters. A partir de la décomposition, nous calculons dans un premier temps une partition des sommets. Ce problème est NP-complet lui-même, nous avons donc utilisé des heuristiques pour le résoudre.

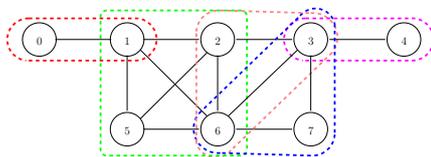


FIG. 1 – Une décomposition arborescente d'un graphe de compatibilité

## 2 Recherche Tabou

Nos heuristiques peuvent trouver des partitions de qualité moyenne, nous avons donc développé une méthode de recherche tabou [2, 3] pour améliorer les résultats.

Etant donné une solution incomplète, vecteur de couples (*objet, cluster*) où certains objets ne sont pas encore affectés à des clusters, la recherche tabou cherche une affectation de ces objets de manière à minimiser le nombre de conteneurs obtenus en traitant les différents clusters.

À partir d'une solution courante, la recherche tabou effectue des mouvements dans l'espace des solutions possibles afin de trouver de meilleures solutions. Dans notre cas, le mouvement choisi est celui qui apportera la meilleure amélioration à la fonction objectif définie en terme de nombre de conteneurs.

Un mouvement qui consiste à affecter un objet à un cluster est dit mouvement *constructif* et *destructif* s'il s'agit d'enlever un objet d'un cluster. L'existence de ces deux types de mouvements influence la recherche tabou et lui permet d'osciller entre des phases de *construction* et de *destruction*. L'alternance des deux phases joue le rôle de diversification qui permet à la recherche tabou de s'échapper des minima locaux et de visiter de nouvelles zones prometteuses.

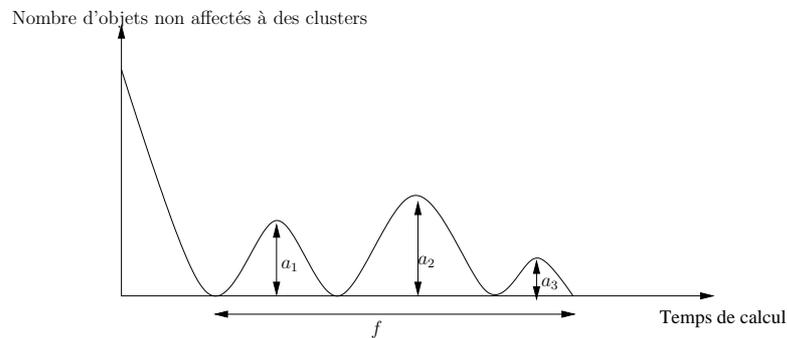


FIG. 2 – Comportement de la recherche tabou dans l'espace de recherche. La fréquence  $f$  est le nombre de solutions complètes à atteindre. L'amplitude  $a$ , générée aléatoirement, représente le nombre d'objets qui peuvent être enlevés de leurs clusters au cours d'une phase de destruction.

Une phase d'intensification est aussi mise en place et lancée à chaque fois qu'une solution complète est atteinte au cours de la recherche. Elle consiste à appliquer une heuristique d'amélioration sur la solution, qui vide certains conteneurs en distribuant leurs objets dans les autres conteneurs.

### 3 Conclusion

Les résultats que nous avons obtenus montrent l'intérêt de la décomposition arborescente pour la résolution de ce problème. En utilisant des heuristiques simples pour calculer la partition des objets, on améliore les résultats obtenus par les heuristiques classiques utilisées pour résoudre le problème. L'utilisation de la recherche tabou permet d'améliorer sensiblement les résultats.

### Références

- [1] K. Jansen and S. R. Öhring. Approximation algorithms for time constrained scheduling. *Information and Computation*, 132:85–108, 1997.
- [2] F. Glover. Tabu search - part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [3] F. Glover. Tabu search - part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.
- [4] M. R. Garey E. G. Coffman and D. S. Johnson. Approximation algorithms for bin-packing – an updated survey. *Algorithm design for computer system design*, Vienna : Springer, 49–106, (1984).