Branchement local pour le problème de sac-à-dos multidimensionnel à choix multiples

Nawal Cherfi[1] et Mhand Hifi[1,2]

1. CERMSEM, MSE, Université Paris 1, 106-112, Bd de l'Hôpital, 75013 Paris, France.
Nawal.Cherfi@malix.univ-paris1.fr

2. LaRIA, Université de Picardie Jules Verne, 5 rue du Moulin Neuf, 80000 Amiens, France.
hifi@laria.u-picardie.fr

Mots-clefs : problème de sac à dos, génération de colonnes, optimisation, branchement local.

1 Introduction

De nombreux problèmes d'optimisation combinatoire issus des applications réelles se modélisent comme des problèmes linéaires en nombres entiers (Wolsey [8]) ou des problèmes linéaires mixtes. La plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles, ce qui complique leur résolution de manière exacte et laisse place aux méthodes de résolution approchée par des heuristiques ou des méthaheuristiques.

Dans cet exposé, nous nous intéresserons à la résolution approchée du problème de sac-à-dos généralisé à choix multiple (noté MMKP) qui est une version plus générale du fameux sac-à-dos unidimensionnel (Balas and Zemel [2]). Le MMKP a plusieurs applications pratiques, comme l'allocation de ressources dans un système multimédia avec de multiples sessions (Akbar *et al.* [1], Hifi *et al.* [4], Khan [6], Moser *et al.* [7]). Une instance du problème est caractérisée par la donnée d'un vecteur capacité $R = (R^1, ..., R^k, ..., R^m)$, et d'un ensemble $J = (J_1, ..., J_i, ..., J_n)$ de variables divisées en n classes disjointes, où chaque classe $J_i,\ i = 1, \ldots, n$, est composée d'un nombre fini d'objets $l_i = |J_i|$. Chaque objet j appartenant à une classe J_i est représenté par un profit v_{ij} et par un vecteur poids $r = (r_{ij}^1, ...r_{ij}^k..., r_{ij}^m)$. Toutes les valeurs v_{ij} et r_{ij}^k sont positives.

Le but du MMKP consiste à attribuer au sac exactement un seul élément de chaque classe avec le maximum de profit sans violer les contraintes de capacités. Formellement, le MMKP peut être décrit sous la forme suivante :

$$
(\text{MMKP}) \left\{ \begin{array}{lllll}
\text{Maximiser} & U(x) = & \sum_{i=1}^{n} \sum_{j=1}^{l_i} v_{ij} x_{ij} & & \\
\text{s.c.} & & \sum_{i=1}^{n} \sum_{j=1}^{l_i} x_{ij} r_{ij}^k \leq R^k & k = 1, \ldots, m & (1) \\
& & \sum_{j=1}^{l_i} x_{ij} = 1 & i = 1, \ldots, n & (2) \\
& & x_{ij} \in \{0, 1\} & i = 1, \ldots, n,\ j = 1, \ldots, l_i. &
\end{array} \right.
$$

D'un point de vue expérimental, la formulation originale du MMKP n'est pas souhaitable à utiliser ; spécialement, pour des instances du problème de taille large. En effet, le nombre des variables et des contraintes s'accroît rapidement suivant, respectivement, l'augmentation du nombre de classes et du nombre des objets de chacune d'elles. Ceci nous a ramené à introduire la technique de génération de colonnes dans notre méthode de résolution, notée (PA), qui a pour idée principale de faire coopérer une méthode d'arrondi et un algorithme exact. Elle s'appuie sur deux phases :

(i) La première phase consiste à appliquer, de manière itérative, la génération de colonnes et arrondir la solution optimale de la relaxation en continu obtenue pour fixer une partie des variables

(ii) La deuxième phase est appliquée pour résoudre de façon exacte la partie restreinte du problème.

Par la suite, nous combinons la procédure (PA) avec la méthode de branchement local. Cette dernière est une méthode de recherche par voisinage, proposée par Fischetti et Lodi dans [3] où les voisinages sont obtenus en introduisant des inégualités linéaires non-valides. La linéarité de ces contraintes ajoutées au problème offre la possibilité d'explorer les voisinages déduits en utilisant des solveurs commerciaux tel que "Cplex". À l'origine le branchement local visait à améliorer l'aspect heuristique des solveurs des problèmes linéaires mixtes.
Dans notre technique de résolution nous introduisons la procédure (PA) dans un schéma de branchement local qu'on notera (BL), où elle sera utilisée comme boite noire pour explorer les voisinages (fournir les solutions).

Enfin, nous présentons une version augmentée de l'aproche (BL) dont l'idée essentielle est d'imiter une méthode de Branch and Bound (B&B) mais sans garantir l'optimalité des solutions obtenues. Cette version de l'approche consiste globalement à appliquer l'heuristique (BL) proposée à un ensemble de nœuds séléctionés d'un arbre de branch-and-bound trancaturé.

2 Méthodologie

Dans cet exposé, nous proposons un algorithme de branch-and-bound trancaturé dans lequel certains nœuds de l'arbre développé sont résolus en utilisant la procédure de branchement local (BL). L'approche proposée peut se résumer dans les étapes suivantes :

1. Démarrer avec une solution réalisable.
2. Générer un ensemble de nœuds selon les critères de la séparation dans un B&B.
3. Appliquer l'heuristique (BL) à un ensemble de nœuds élites (par séléction).

L'heuristique que nous utilisons est basée sur la méthode du branchement local dans lequel une procédure d'arrondi combinée avec la génération de colonnes est utilisée pour produire des solutions réalisables. Elle peut se résumer dans les étapes suivantes :

1. Soit η un nœud actif en cours d'exploration contenant un programme linéaire restreint (ILP) associé au MMKP.
2. Soit $\bar{x}$ une solution réalisable de (ILP) tel que $\bar{S} = \{j \in S : \bar{x_j} = 1\}$ alors effectuer une partition de l'espace de recherche en deux disjonctions :
$\sum_{j\in\bar{S}}(1-x_j) + \sum_{j\in N\setminus\bar{S}} x_j \leq k$ (nœud gauche) ou $\sum_{j\in\bar{S}}(1-x_j) + \sum_{j\in N\setminus\bar{S}} x_j \geq k+1$ (nœud droit)
3. Appliquer la procédure de génération de colonnes au problème (ILP) du nœud gauche.
4. Appliquer une procédure d'arrondi au problème (ILP) pour fixer une partie des variables, notée I1, et compléter la deuxième partie, notée I2, en utilisant un algorithme exact. Soit $\bar{x_2}$ la solution obtenue.
5. Mettre à jour la solution obtenue :
   Si $(U(\bar{x_2}) > U(\bar{x}))$ **alors**
   $\bar{x} \longleftarrow \bar{x_2}$, répéter les étapes 2-5 jusqu'à la vérification du test d'arrêt.
   sinon utiliser une technique de diverstification.

6. Créer de nouveaux nœuds en utilisant des stratégies spéciales de branchement et supprimer les nœuds dont la valeur objectif est inférieure ou égale à la meilleure solution obtenue.
7. Sortir avec la meilleure solution si le test d'arrêt est vérifié ou lorsque la liste des nœuds est vide ; répéter les étapes 2-6 dans le cas contraire.

La méthode est ensuite testée sur des instances de littérature et comparée aux résultats du solveur "Cplex" et une méthode de recherche locale réactive [4]. L'approche proposée améliore les résultats de la majorité de ces instances en un temps de calcul raisonable.

Références

[1] M. M. Akbar, M. W. H. Sadid, M. S. Rahman, M. A. H. Newton and M. Kaykobad. A Parallel Heuristic Algorithm for Multiple-Choice Multi-Dimensional Knapsack Problem, *Working Paper,* Department of CSE, BUET, Dhaka-1000, Bangladesh, 2005.

[2] E. Balas and E. Zemel. An algorithm for large zero-one knapsack problem, *Operations Research, vol. 28, pp.1130-1154, 1980.*

[3] M.Fischetti and A.Lodi.'Local branching', *Mathematical Programming, Vol. 98, pp.23–47, 2003*

[4] M. Hifi, M. Michrafy and A. Sbihi, A reactive local search-based algorithm for the mutiple-choice multi-dimensional knapsack problem. *Computational Optimization and Applications,* to appear.

[5] H. Kellerer, U. Pferschy, D. Pisinger, *Knapsack Problems,* Springer Verlag, ISBN 3- 540-40286-1.

[6] S. Khan, Quality adaptation in a multi-session adaptive multimedia system : model and architecture, *PhD Thesis,* Department of Electronical and Computer Engineering, Uiversity of Victoria, May 1998.

[7] M. Moser, D. P. Jokanović and N. Shiratori. An algorithm for the multidimesional multiple-choice knapsack problem, *IEECE Transactions on Fundamentals of Electronics, vol. 80, No 3., pp.582-589, 1997.*

[8] L. A. Wolsey, *Integer Programming,* Wiley-Interscience, 1998.