Recherche à voisinage variable pour le sac-à-dos avec remises

Christophe Wilbaut[1,2], Raca Todosijevic[1,2], Saïd Hanafi[1,2], Arnaud Fréville[1 a]

[1] Univ. Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201, F-59313 Valenciennes, France

[2] INSA Hauts-de-France, F-59313 Valenciennes, France

{christophe.wilbaut,raca.todosijevic,said.hanafi}@uphf.fr

Mots-clés : *Sac-à-dos, Métaheuristique, Recherche à voisinage variable, Fixation*

Le problème du sac à dos (SAD) a été étudié intensivement depuis le milieu du vingtième siècle et de nombreuses variantes difficiles, obtenues en ajoutant des contraintes et/ou changeant l'objectif par exemple, ont depuis émergé avec d'importantes applications [3, 4]. Le problème du sac-à-dos avec remises (SADR) est une extension introduite en 2007 [1]. Dans la terminologie anglaise le mot *discounted* est utilisé pour faire un parallèle avec les remises promotionnelles pratiqués dans le commerce : de façon similaire à l'achat simultané de deux articles pendant la saison des promotions qui génère une promotion, la sélection d'un objet dans le SADR implique une sorte de remise. Plus précisément les objets sont regroupés par trois et le profit de l'objet avec remise est la somme des profits des deux autres objets du groupe. De plus, l'objet en promotion consomme moins de ressources que le cumul des deux autres objets. Formellement le SADR implique un ensemble M de m groupes de trois objets tels que $N_i = \{3i, 3i+1, 3i+2\}$ pour $i \in M = \{0, \ldots, m-1\}$, pouvant être ajoutés dans un sac de capacité b. Chaque objet $3i+k$ ($k \in \{0,1,2\}$) du groupe N_i ($i \in M$) est caractérisé par un profit c_{3i+k} et un poids a_{3i+k} tels que : $c_{3i+2} = c_{3i} + c_{3i+1}$ et $\max\{a_{3i}, a_{3i+1}\} < a_{3i+2} < a_{3i} + a_{3i+1}$. Comme dans le SAD toutes les données sont supposées entières et positives. L'objectif du SADR est de choisir au plus un objet par groupe afin de maximiser le profit total en respectant la contrainte de sac-à-dos. Le SADR peut être formulé comme une extension du SAD en introduisant les variables binaires x_{3i}, x_{3i+1} et x_{3i+2} pour la sélection des objets dans le groupe N_i :

$$\max \sum_{i \in M} c_{3i}x_{3i} + c_{3i+1}x_{3i+1} + c_{3i+2}x_{3i+2} \tag{1}$$

$$\text{s.c. : } \sum_{i \in M} a_{3i}x_{3i} + a_{3i+1}x_{3i+1} + a_{3i+2}x_{3i+2} \leq b \tag{2}$$

$$x_{3i} + x_{3i+1} + x_{3i+2} \leq 1 \qquad i \in M \tag{3}$$

$$x_{3i}, x_{3i+1}, x_{3i+2} \in \{0,1\} \qquad i \in M \tag{4}$$

La fonction objective (1) vise à maximiser le profit total des objets sélectionnés. La contrainte (2) est la contrainte de sac-à-dos, alors que les contraintes (3) oblige le choix d'au plus un objet par groupe. Notons que le SADR peut aussi être vu comme un cas particulier du sac-à-dos à choix multiples [7]. Plusieurs travaux basés sur des métaheuristiques, notamment inspirées de phénomènes naturels, ont été appliquées ces dernières années pour résoudre le SADR [2, 6, 8, 9].

Dans ce travail nous proposons une recherche à voisinage variable (RVV) [5]. De nombreuses variantes de la RVV ont été proposées depuis, dont la RVV générale (RVVG) que nous mettons en œuvre ici. Notre méthode repose notamment sur l'utilisation de quatre voisinages définis à partir des notations suivantes :

— $|x|$ - cardinalité de la solution x, i.e., nombre d'objets sélectionnés, $|x| = \sum_{i \in M} \sum_{j \in N_i} x_j$.

— $d_{inter}(x, x')$ - distance entre les solutions x et x' en nombre de groupes différents contribuant à la solution, i.e., $d_{inter}(x, x') = \sum_{i \in M} |\sum_{j \in N_i}(x_j - x'_j)|$.

[a]Professeur émérite - retraité.

— $d_{intra}(x, x')$ - distance entre les solutions x et x' calculée comme la somme des distances de Hamming dans les groupes d'objets, i.e., $d_{intra}(x, x') = \sum_{i \in M} \sum_{j \in N_i} |x_j - x'_j|$.

Les quatre voisinages utilisés sont alors définis comme suit :

1. $V_1(x)$ = `Ajout(x)` $= \{x' : |x'| = |x| + 1\}$: inclut toutes les solutions qui peuvent être dérivées de x en ajoutant exactement un objet.
2. $V_2(x)$ = `Retire(x)` $= \{x' : |x'| = |x| - 1\}$: idem en retirant exactement un objet.
3. $V_3(x)$ = `Inter(x)` $= \{x' : d_{inter}(x, x') = 2, d_{intra}(x, x') \leq 1\}$: inclut toutes les solutions qui peuvent être dérivées à partir de x en remplaçant un objet par un autre d'un autre groupe.
4. $V_4(x)$ = `Intra(x)` $= \{x' : d_{intra}(x, x') = 2\}$: même principe au sein du même groupe.

La RVVG proposée utilise ainsi les voisinages V_2 à V_4 dans les phases d'intensification de type descente à voisinage variable, et le voisinage V_1 dans la procédure de perturbation.

Nous avons utilisé plusieurs heuristiques pour initialiser la RVVG et combiner celle-ci avec des techniques de fixation que nous ne pouvons présenter ici faute de place. L'approche est validée sur 80 instances de la littérature et les résultats sont présentés succinctement dans le Tableau 1 où la ligne `#Égalité` (resp. `#RVV`, `#Lit.`) donnent le nombre de fois que notre approche a retourné une solution ayant la même (resp. une meilleure, une moins bonne) valeur que les travaux de la littérature ([6] et [9] ne considèrent que 30 et 40 des 80 instances, respectivement).

	[2]	[8]	[6]	[9]
#Égalité	14	11	3	5
#**RVV**	30	50	25	26
#Lit.	36	19	2	9

TAB. 1 – Comparaison avec quelques métaheuristiques de la littérature

Les valeurs du Tableau 1 montrent la compétitivité de la méthode qui converge très rapidement vers des solutions de très bonne qualité. Les résultats seront détaillés lors de la présentation.

Références

[1] B. Guldan. *Heuristic and Exact Algorithms for Discounted Knapsack Problems.* Master thesis, University of Erlangen–Nürnberg, Germany, 2007.

[2] Y.-C. He, X.-Z. Wang, Y.-L. He, S.-L. Zhao et W.-B. Li. Exact and approximate algorithms for discounted {0-1} knapsack problem. *Information Sciences*, 369:634–647, 2016.

[3] H. Kellerer, U. Pferschy et D. Pisinger. *Knapsack Problems.* Springer Berlin Heidelberg, Berlin, 2004.

[4] J. H. Lorie et L. J. Savage. Three Problems in Rationing Capital. *The Journal of Business*, 28(4):229–239, 1955.

[5] N. Mladenović et P. Hansen. Variable neighborhood search. *Computers Operations Research*, 24 (11): 1097–1100, 1997.

[6] T. K. Truong. A new moth-flame optimization algorithm for discounted 0-1 knapsack problem. *Mathematical Problems in Engineering*, 21, 2021.

[7] C. Wilbaut, S. Hanafi et S. Salhi. A survey of effective heuristics and their application to a variety of knapsack problems. *IMA Journal of Management Mathematics*, 19(3):227–244, 2007.

[8] C. Wu, J. Zhao, Y. Feng et M. Lee. Solving discounted {0-1} knapsack problems by a discrete hybrid teaching-learning-based optimization algorithm. *Applied Intelligence*, 50: 1872–1888, 2020.

[9] H. Zhu, Y. He, X. Wang et E.C.C. Tsang. Discrete differential evolutions for the discounted {0–1} knapsack problem. *International Journal of Bio-Inspired Computation*, 10(4):2019–238, 2017.