# Problème du sac-à-dos disjonctif : résolution par programmation dynamique

Siao-Leu Phouratsamay[1,2], François Clautiaux[1,2], Pierre Pesneau[1,2]

[1] Université de Bordeaux, Institut de Mathématiques de Bordeaux, UMR 5251

[2] INRIA Bordeaux - Sud-Ouest, France

{siao-leu.phouratsamay, francois.clautiaux, pierre.pesneau}@u-bordeaux.fr

**Mots-clés** : *Sac-à-dos disjonctif, programmation mathématique, programmation dynamique.*

## 1 Introduction

Le problème du sac-à-dos disjonctif est une extension du problème du sac-à-dos où il existe des conflits entre les objets. Soit $V = \{1, \dots, n\}$ l'ensemble des $n$ objets. Chaque objet $i \in V$ a un poids positif $w_i$ et un profit positif $p_i$. La capacité du sac est notée $W$ où $W$ est telle que $\sum_{i \in V} w_i > W$. Deux objets $i$ et $j$ sont en conflit s'ils ne peuvent pas être sélectionnés ensemble dans le sac. Les conflits entre les objets sont représentés par un graphe de conflits non orienté $G = (V, E)$ où $E$ est l'ensemble des arêtes représentant les conflits entre les objets, *i.e.* une arête $(i, j) \in E$ existe si les objets $i$ et $j$ sont en conflit. Le but est de déterminer un sous-ensemble d'objets maximisant le profit du sac et satisfaisant les contraintes de capacité et de conflits. Le problème est clairement NP-difficile, puisqu'il généralise le problème de stable de poids maximum dans un graphe.

Soit $x_i$ la variable de décision binaire valant 1 si l'objet $i$ est sélectionné dans le sac, et 0 sinon. La formulation mathématique du problème du sac-à-dos disjonctif est définie par :

$$\max \sum_{i \in V} p_i x_i \tag{1}$$

$$\text{s.t.} \sum_{i \in V} w_i x_i \leq W, \tag{2}$$

$$x_i + x_j \leq 1 \qquad \forall (i, j) \in E, \tag{3}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V. \tag{4}$$

La fonction objectif (1) consiste à maximiser le profit des articles sélectionnés. La contrainte (2) assure que la capacité du sac n'est pas dépassée. Les contraintes (3) interdisent l'objet $i$ d'être choisi avec l'objet $j$ s'ils sont en conflits.

## 2 État de l'art

Le problème du sac-à-dos disjonctif a été introduit par Yamada *et al.* [6]. Ils ont développé un algorithme de branchement utilisant une borne supérieure obtenue par relaxation lagrangienne des constraintes de conflits. Leur algorithme a été testé sur des graphes de conflits peu denses. Le meilleur algorithme de branchement existant a été proposé récemment par Bettinelli *et al.* [1]. Les auteurs ont déterminé une nouvelle borne supérieure en exploitant la borne de couverture par des cliques pondérées proposée par [2] pour l'étendre au cas avec capacité. Leur expérimentation sur des graphes de conflits plus ou moins denses ont montré que leur algorithme de branchement est efficace pour les graphes dont la densité est supérieure à 10%.

Des algorithmes de programmation dynamique pseudo-polynomiaux existent pour des graphes de conflits spécifiques [4, 5]. En particulier, Sadykov *et al.* [5] ont mis en place un algorithme de programmation dynamique en $\mathcal{O}(nW)$ pour le cas où le graphe de conflit est un graphe d'intervalle.

Les travaux de Ben Salem *et al.* [1] sont les seules études polyhédrales trouvées dans la littérature pour le problèmes du sac-à-dos disjonctif. Ils ont étudié la formulation (1)-(4) et ont proposé de nouvelles inégalités valides en lien avec les inégalités de clique et de couverture.

## 3 Approche étudiée

Résoudre le problème avec un graphe de conflits quelconque en utilisant un algorithme de programmation dynamique se fait en temps exponentiel. Ceci est dû au fait que l'on doit conserver dans l'espace d'état la trace des articles déjà sélectionnés afin d'éviter de sélectionner des paires d'articles en conflit.

Notre approche consiste à déterminer des stratégies pour obtenir une approche compétitive basée sur la programmation dynamique pour résoudre le problème de sac à dos disjonctif. Nos algorithmes reposent sur la relaxation lagrangienne, des techniques d'agrégation de contraintes et de précalculs de bornes.

## Références


[1] Andrea Bettinelli, Valentina Cacchiani, and Enrico Malaguti. A branch-and-bound algorithm for the knapsack problem with conflict graph. *INFORMS J. on Computing*, 29(3) :377–580, 2017.

[2] S Held, W Cook, and EC Sewell. Maximum-weight stable sets and safe lower bounds for graph coloring. *Mathematical Programming Computation*, 4(4) :363–381, 2012.

[3] Mhand Hifi and Mustapha Michrafy. Reduction strategies and exact algorithms for the disjunctively knapsack problem. *Computers & Operations Research*, 34 :2657–2673, 09 2007.

[4] Ulrich Pferschy and Joachim Schauer. The knapsack problem with conflict graphs. *Journal of Graph Algorithms and Applications*, 13(2) :233–249, 2009.

[5] Ruslan Sadykov and François Vanderbeck. Bin packing with conflicts : A generic branch-and-price algorithm. *INFORMS J. on Computing*, 25(2) :244–255, April 2013.

[6] T. Yamada, S. Kataoka, and K. Watanabe. Heuristic and exact algorithms for the disjunctively constrained knapsack problem. *IPSJ Journal*, 43(9) :2864–2870, 2002.