Heuristiques pour des problèmes de découpe guillotine par batchs

Quentin Viaud[1,2], François Clautiaux[1,2], Ruslan Sadykov[2,1], François Vanderbeck[1,2]

[1] IMB (UMR CNRS 5251), Université de Bordeaux
351 cours de la Libération, 33400 Talence Cedex, France
[2] INRIA Bordeaux - Sud-Ouest
351 cours de la Liberation, 33405 Talence, France
`quentin.viaud@u-bordeaux.fr, francois.clautiaux@math.u-bordeaux.fr,`
`ruslan.sadykov@inria.fr, fv@math.u-bordeaux.fr`

Mots-clés : *Programmation mathématique, Bin-packing en deux dimensions, Découpe guillotine, Décomposition de Dantzig-Wolfe, Heuristique de diving*

1 Introduction

Le problème de bin-packing en deux dimensions est un problème classique de la littérature, qui vise à découper un ensemble d'articles $\mathcal{I}$ dans des bins rectangulaires identiques de taille (W, H). L'objectif est de minimiser le nombre de bins. Chaque article $i \in \mathcal{I}$ doit être coupé exactement d_i fois et est de dimension (w_i, h_i). Nous nous intéressons ici au cas "guillotine", où une coupe doit s'effectuer en ligne droite d'un bord à l'autre d'une plaque et plus particulièrement à la version en quatre niveaux restreinte. Chaque article doit être obtenu en au plus quatre coupes guillotine. Nous nous limitons ici aux configurations de coupe telles que la distance entre deux coupes successives correspond à la hauteur ou longueur d'un article. La rotation à 90 degrés des articles est autorisée.

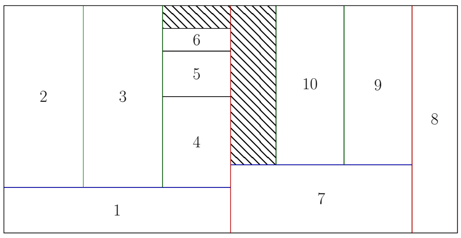

FIG. 1 – Représentation d'un plan de découpe en 4 niveaux pour une plaque

La spécificité de notre problème est le fait que l'ensemble des articles à découper est composé de k lots successifs $\mathcal{I}_1, \ldots, \mathcal{I}_k$ devant être coupés à la suite. Pour des raisons pratiques liées à l'entreposage du matériel, seul le dernier bin utilisé pour couper les articles du batch $\mathcal{I}_\ell$ peut être réutilisé pour découper des articles du batch $\mathcal{I}_{\ell+1}$ ($\ell = 1, \ldots, k-1$).

Pour des instances réelles et dans le cadre des modèles existants, la résolution du problème global est hors de portée des méthodes basées sur la programmation mathématique. La stratégie que nous utilisons est donc de résoudre successivement les problèmes liés à chacun des batchs, en gardant le dernier bin utilisé par le batch précédent.

La question qui se pose est la fonction objectif à utiliser pour le dernier bin de chaque batch. Nous avons comparés plusieurs approches : minimiser la longueur des coupes de niveau un du dernier bin, ou chercher à ranger le maximum d'articles du batch suivant dans ce dernier bin. Ces deux stratégies sont comparées numériquement en terme de nombre de bins utilisés au total.

2 Méthodologie générale

L'approche classique de résolution du problème de bin-packing est de le reformuler en utilisant la décomposition de Dantzig-Wolfe [3] (voir par exemple [4]). Celle-ci décompose le problème initial en un problème maître (de type set partitionning) et un sous-problème de sac-à-dos guillotine en quatre niveaux restreint avec rotation des articles. Le principe est le même pour toutes les fonctions objectif considérées dans ce travail.

Dans [2], nous avons montré comment le sous-problème pouvait être résolu de manière exacte. Dans le cadre de la génération de colonnes, les temps de calcul sont trop importants. Nous relâchons donc les contraintes de demande des articles afin d'obtenir un programme dynamique de taille raisonnable (voir [2]). Les colonnes générées peuvent donc contenir un nombre trop important d'articles d'un certain type (colonnes non réalisables). Cependant, les valeurs duales associées aux articles permettent de ne pas trop détériorer la qualité de la relaxation linéaire du maître [1].

Le problème maître entier est résolu à l'aide d'une heuristique de diving. Celle-ci vise à sélectionner une colonne réalisable (respectant les contraintes de demande des articles) dans la solution du maître et de la fixer à une valeur entière. Le maître est alors mis à jour (réduction des contraintes de demande des articles) et une génération de colonne recommence. Si la solution du maître ne contient aucune colonne réalisable, nous utilisons une heuristique simple pour le sous-problème afin d'en fournir une.

3 Synthèse des résultats expérimentaux

Afin de déterminer quelle est la meilleure fonction objectif, nous avons lancé des expérimentations pour calculer le nombre de bins à utiliser pour traiter dix batchs d'articles. Le problème est résolu avec pour objectif de minimiser le nombre de bins (MB), de minimiser la longueur du dernier bin ($MLBL$), et de minimiser la surface perdue (MUA). Dans le cadre d'une résolution $MLBL$, le résiduel du dernier bin est utilisé pour le batch suivant. Dans le cadre d'une résolution MUA, le nombre de bins pouvant contenir des articles du batch courant et du batch suivant est fixé à un.

Les premiers résultats montrent que les fonctions objectif $MLBL$ et MUA permet de réduire le nombre de bins à utiliser pour plusieurs batchs de manière significative. Ces résultats sont obtenus au prix d'un temps de calcul significativement plus grand que pour l'objectif simple MB.

Références

[1] G.F. Cintra, F.K. Miyazawa, Y. Wakabayashi, and E.C. Xavier. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191(1) :61 – 85, 2008.

[2] François Clautiaux, Ruslan Sadykov, François Vanderbeck, and Quentin Viaud. Méthodes de résolution pour un problème de sac-à-dos en deux dimensions. In *Conférence ROADEF - Compiègne*, 2016.

[3] George B. Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1) :101–111, 1960.

[4] François Vanderbeck. A nested decomposition approach to a three-stage, two-dimensional cutting-stock problem. *Management Science*, 47(6) :864–879, 2001.