

Un algorithme hybride pour le problème de découpe de verre de Saint-Gobain

Michel Vasquez, Mirsad Buljubasic

Ecole des Mines d'Ales, LGI2P, France

{michel.vasquez,mirsad.buljubasic}@mines-ales.fr

Mots-clés : *algorithme glouton, recherche locale, énumération, découpe de verre*

1 Introduction

La société Saint-Gobain, principal fabricant de verre plat dans le monde, a proposé pour le challenge ROADEF/EURO 2018 un problème de découpe 2D avec contrainte de guillotine. Il s'agit de produire un ensemble de *petits* rectangles (commande client à honorer impérativement) à partir de grandes plaques de verre de dimension constante (6000×3210 cm) tout en minimisant la surface de verre perdu. C'est un problème de binpacking 2D particulier : les grandes plaques de verre sont assimilées à des *bins*, les *petits* rectangles à des *items*. Trois niveaux de découpes sont autorisés : le premier niveau correspond à une découpe verticale **1-cut**, le deuxième niveau est une découpe horizontale **2-cut** dans le rectangle de verre produit au niveau précédent, le troisième niveau **3-cut** est une découpe verticale supplémentaire. Une ultime **4-cut** est autorisée pour produire un *item*. Ces niveaux sont illustrés dans la figure ci-dessous.

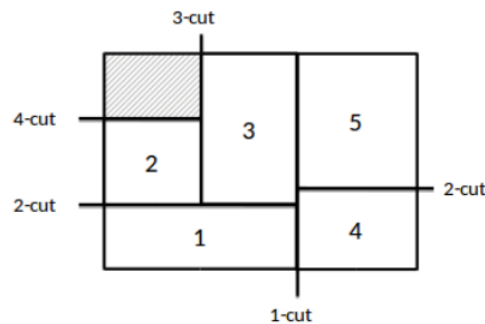


FIG. 1 – Exemple de découpe réalisable

Les approches proposées dans la littérature ([1, 2]), sont difficilement adaptables au problème de Saint-Gobain à cause de quelques caractéristiques liées au processus industriel de fabrication de ces plaques de verre. Une des principales spécificités est la prise en compte de défauts présents dans les grandes plaques de verre (*bin*) servant de matrice pour la production des *items*. Tout d'abord aucun défaut ne doit être présent à l'intérieur de ces derniers. Ensuite aucune découpe guillotine ne doit traverser un défaut. Enfin la présence de ces défauts rend les *bins* pour la plupart différents et induit un ordre de passage de ces derniers qui durcit la contrainte sur l'ordre de production des *items*. Des contraintes d'espacement minimal entre découpes parallèles viennent également compliquer le placement des rectangles dans les grandes plaques de verre. Les découpes d'*items* s'effectuent de la gauche vers la droite. La fonction objective consiste, pour une ensemble d'*items* donné, à minimiser le nombre de *bins* utilisés et à maximiser la surface à droite de la dernière découpe guillotine dans le dernier *bin*.

2 Une approche hybride

Nous considérons ce problème sous les deux angles suivants : (1) définir les sous-ensembles d'*items* affectés à chaque *bins*; (2) positionner les découpes pour produire ces *items* en respectant toutes les contraintes. L'approche générale est un algorithme glouton qui remplit les *bins* un par un. Pour chaque *bin*, l'objectif est de maximiser la somme des surfaces des items qui y sont placés. Plus précisément, nous remplissons le premier *bin* le plus possible puis nous passons au *bin* suivant et ce jusqu'à ce que le dernier *item* soit placé. Le sous-problème de remplissage de chaque *bin* est traité par une approche qui combine un second glouton, une recherche tabou et une recherche exhaustive. Nous essayons de placer le plus possible d'*items* d'une séquence d'*items* donnée dans le *bin* courant (sans changer leur ordre) en définissant des 1-cut une par une. Une recherche arborescente, en profondeur d'abord, est utilisée pour définir la 1-cut qui minimise localement la surface de verre perdu. La combinatoire de cette procédure d'énumération prend en compte la définition des 2-cut, 3-cut et 4-cut ainsi que l'orientation des *items* dans ces *cuts*. La recherche d'un ordre des *items* qui permette, par l'usage de ces deux procédures, de remplir au maximum le *bin* courant est réalisée par une recherche locale qui explore l'espace des séquences en échangeant la position d'une paire d'*items non tabou* à chaque itération.

3 Résultats numériques

L'expérimentation a porté sur des instances d'une centaine d'*items* mises à disposition par Saint-Gobain. La table ci-dessous reporte les résultats obtenus lors de la compétition en une heure de temps cpu. Les colonnes **Waste** et **%** représentent la surface totale et le pourcentage de verre perdu. On compare nos résultats avec les valeurs des meilleures solutions trouvées par les équipes en compétition.

Instance	Notre algorithme		Meilleures valeurs	
	Waste	%	Waste	%
B1	2722308	3.48	2661318	3.4
B2	15137885	4.38	13674125	3.96
B3	18191093	4.64	18191093	4.64
B4	9857995	5.64	8269045	4.73
B5	72155615	18.41	72155615	18.41
B6	11407117	5.4	11195257	5.3
B7	8891889	4.36	8355819	4.1
B8	16067959	4.05	16067959	4.05
B9	17484577	5.38	17484577	5.38
B10	24095813	6.25	21951533	5.69
B11	22584380	5.73	22584380	5.73
B12	15903967	5.6	13958707	4.92
B13	26634915	5.26	24471375	4.83
B14	9449200	4.93	8656330	4.52
B15	26616371	5.11	24517031	4.71
moyenne	19813405.6	5.91	18946277.6	5.62

TAB. 1 – Résultats obtenus sur le *dataset B*.

Références

- [1] Jakob Puchinger and Gunther Raidl. *Models and algorithms for three-stage two-dimensional bin packing*. European Journal of Operational Research, 127(3), 1304–1327, 2007.
- [2] Victor Parada, Mauricio Sepúlveda, Mauricio Solar, Arlindo Gomes. *Solution for the constrained guillotine cutting problem by simulated annealing*. Computers & Operations Research, 25(1), 37–47, 1998.