

Problème d'Assemblage Orthogonal Rectangulaire, Approche Algorithmique

Rachid Ouafi¹ et Isma Dahmani¹

1Département de Recherche Opérationnelle
Faculté de Mathématiques, BP 32
EL Alia Bab Ezzouar, 16111, Alger
ALGERIE.
dahmani.isma@gmail.com
rachid_ouafi@hotmail.com

RÉSUMÉ. Le problème traité dans cet article est le problème d'assemblage orthogonal rectangulaire. Il consiste à regrouper un ensemble de pièces rectangulaires dans un rectangle final de surface minimale. Dans cette étude, une contrainte est imposée, à savoir l'orientation fixe des pièces. Nous proposons dans cet article, une version modifiée de la méthode exacte BFBB (Best First Branch and Bound) [8]. En vue d'améliorer la méthode en temps d'exécution et en place mémoire. A cet effet nous utilisons trois stratégies. Nous commençons par le développement d'une nouvelle borne supérieure afin de réduire l'espace de recherche. Ensuite, nous introduisons une nouvelle représentation [2] de la liste fermée de l'algorithme pour une meilleure réorganisation. Enfin, la notion d'ordre lexicographique est utilisée afin d'éliminer les modèles de découpe dupliqués[9]. La performance de notre algorithme a été évaluée sur un ensemble d'instances générées aléatoirement et les résultats obtenus sont comparés à l'algorithme BFBB.

Mots-clés: Séparation et évaluation, optimisation combinatoire, assemblage rectangulaire.

1 Introduction

Les problèmes de Découpe et d'Assemblage (D&A) ont été posés dès les années 60 (Kantorovich, 1960 ; Dyckhoff, 1990; Hifi, 1994, G. Wäscher, H. et al., 2004). De leur intérêt, plusieurs axes de recherche ont suscités l'attention de beaucoup de chercheurs et divers papiers ont fait leur apparition. Le problème de placement (découpe) consiste à placer (découper) un ensemble de pièces dans un rectangle de largeur et hauteur fixes (*packing problem*) ou largeur fixe et de hauteur non fixe (*strip packing*, Hinxman, 1980; Coffman et Lagarias, 1989; Jacobs, 1996 ; Zhang, Kang , Deng 2006 ; Cui, Y. et al., 2008). Les problèmes SP et RP (*Square Packing* and *Rectangular Packing*) ont été beaucoup étudiés par plusieurs chercheurs (Kleitman et Krieger, 1975; Duijvestijn, 1978; Hifi, 1994, Hifi et Ouafi 1998 ; Clautiaux F. et al., 2006). Le SP (RP) consiste à regrouper des pièces de formes carrées (rectangulaires) dans un carré (rectangle) final de surface minimale.

Nous nous intéressons dans cet article, au problème d'assemblage orthogonal rectangulaire (PAOR). Le PAOR consiste à regrouper des pièces rectangulaires dans un rectangle final de surface minimale. Son dual[12] est un problème de découpe de stock à 2D (Two-Dimensional Cutting Stock Problem(TDCS)). Une contrainte guillotine est imposée au problème TDCS afin de réduire les modèles de découpes faisables. La découpe guillotine consiste à couper un rectangle d'un bord à l'autre. De même cette contrainte apparait dans le cas du PAOR sous forme de construction orthogonale [8]. Elle se fait en combinant les pièces avec leurs copies sous forme de constructions horizontales et verticales (voir figure 1).

Hifi & Ouafi[8] ont proposé une méthode exacte pour la résolution du PAOR appelée Best First Branch and Bound (BFBB). Nous proposons une version modifiée de BFBB basée sur l'introduction de nouvelles techniques permettant d'améliorer la performance de celle ci.

Notre papier est structuré comme suit : dans la deuxième section nous commençons par la présentation du PAOR [8]. Ensuite, dans la troisième section nous développons la nouvelle version de BFBB par : (i) développement d'une nouvelle borne supérieure (heuristique gloutonne), (ii) adaptation de la nouvelle représentation de la liste fermée [2]. (iii) introduction de la stratégie d'élimination des modèles dominés et dupliqués [9].

Enfin, dans la quatrième section, la performance de la nouvelle version est évaluée sur des instances de différentes tailles générées aléatoirement.

2 Problème d'Assemblage Orthogonal Rectangulaire (PAOR)

2.1 Présentation du PAOR

Étant donné un ensemble de n pièces $S = \{(l_1, w_1), (l_2, w_2), \dots, (l_n, w_n)\}$, où l_i (resp. w_i) est la longueur (resp. la largeur) et b_i désigne le nombre de copies de la pièce i , $i = 1, \dots, n$.

Une solution réalisable T est un modèle d'assemblage et elle est représentée par un rectangle qui regroupe l'ensemble des pièces de S avec leurs duplications. Une solution optimale T^* est une solution réalisable de surface minimale.

Nous utilisons une stratégie de construction orthogonale pour réduire les modèles d'assemblage faisables. Elle se fait en combinant les pièces avec leurs copies sous forme de constructions horizontales et verticales. A cet effet, les définitions ci-dessous sont adoptées.

Définition 1. Soient p et p' deux pièces de S de dimensions respectives (l, w) et (l', w') . On appelle $R = (l+l', \max\{w, w'\})$ (resp. $R = (\max\{l, l'\}, w+w')$), un assemblage rectangulaire, noté r-assemblage, obtenu en joignant p et p' par une construction horizontale (resp. verticale), sachant que le nombre d'occurrences d'une pièce donnée dans chaque construction ne doit pas excéder la borne supérieure b_i (voir

Fig.1. ci-dessous). Considérons S_{rem} sous-rectangle restant, de longueur l_{rem} et de largeur w_{rem} , après la combinaison de deux pièces p et p' .

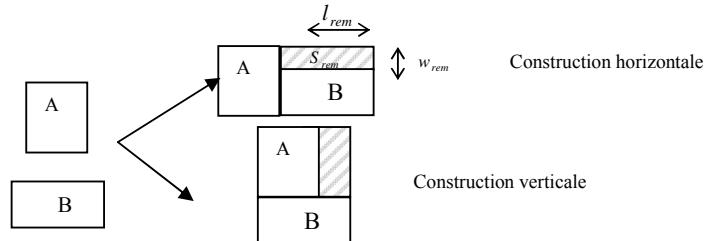


Fig. 1. Constructions horizontale et verticale.

Définition 2. On appelle R un assemblage terminal (noté t-assemblage), s'il contient toutes les pièces avec leurs copies.

Remarque. Dans cette étude, seuls les pièces et les r-assemblages d'orientation fixées sont pris en considération

3 La nouvelle version de l'algorithme BFBB

3.1 La nouvelle borne supérieure

Nous développons un algorithme glouton (Algo BS) pour le calcul de la borne supérieure du PAOR. L'algorithme commence par le calcul de la borne initiale supérieure de départ ($E_{sup}[8]$), qui sera améliorée par la suite. On considère à cet effet une liste ξ contenant l'ensemble des pièces et leurs copies ordonnées selon l'ordre décroissant de largeur. On construit, en premier lieu, l'assemblage R_{rec} constitué uniquement de pièces de largeur maximale. Dans l'étape principale de la procédure, on effectue des constructions horizontales successives de la manière suivante : on introduit une liste intermédiaire ξ' constituée de pièce de largeur maximale parmi les pièces restantes, puis on sélectionne R de ξ' de longueur maximale ; pour la placer à droite de R_{rec} et on effectue par la suite un remplissage avec les pièces restantes sur le sous-rectangle S_{rem} , tant que possible. La procédure s'arrête si $E_{rec} \leq E_{sup}$ tel que E_{rec} représente la surface de R_{rec} et on affecte la valeur de E_{rec} à E_{sup} . Dans ce cas, E_{sup} est la meilleure borne supérieure de départ par construction horizontale. Sinon cette étape est répétée jusqu'à ce que $\xi = \emptyset$.

Algo BS : Une procédure gloutonne pour le PAOR

Entrée: ensemble des pièces rectangulaires borné par b_i , $1 \leq i \leq n$

Sortie: solution sous-optimale notée E_{sup}

L'étape initiale:

- 1) Calculer E_{sup} tel que : $E_{\text{sup}} = \min\{E_h, E_v\}$ où

$$E_h = \left(\sum_{i=1}^n b_i l_i\right) (\max_{1 \leq i \leq n} \{w_i\}) \quad \text{et} \quad E_v = \left(\sum_{i=1}^n b_i w_i\right) (\max_{1 \leq i \leq n} \{l_i\})$$

- 2) ξ : ensemble des pièces avec leurs copies triées selon l'ordre décroissant de leurs largeurs : $w_1 > w_2 \geq \dots \geq w_n$

Soit R_{rec} le rectangle initial obtenu par construction horizontale de la pièce R_1 avec ses copies $R_{\text{rec}} = (b_1 l_1, w_1)$, avec : $\xi = \xi \setminus \{R_i / l_{R_i} = l_{R_1} \text{ et } w_{R_i} = w_{R_1}\}$

L'étape principale:

Répéter

1. Soit ξ' l'ensemble des pièces dont la largeur est la plus grande :

$$\xi' = \{R_k : w_{R_k} = \max_{R_j \in \xi} \{w_{R_j}\}, \forall l_{R_k}\}$$

2. Choisir la pièce : $R' \in \xi' / l_{R'} = \max_{R_k \in \xi'} \{l_{R_k}\}$,
3. faire la construction horizontale de R' avec R_{rec} puis, Poser : $\xi = \xi \setminus \{R'\}$
4. Calculer l'espace vide S_{rem} , résultat de la construction précédente, tel que : $EV = (l_{R'}, w_{R_{\text{rec}}} - w_{R'})$.

Tant que $(EV : (l_{EV}, w_{EV}) \neq (0,0))$ et (il existe une pièce qui rentre dans S_{rem})

Faire

- i. Placer la pièce R_i dans l'espace S_{rem}
- ii. Poser $\xi = \xi \setminus \{R_i\}$
- iii. Calculer le nouvel espace vide S_{rem}

Fait;

Jusqu'à: $(\xi = \emptyset)$ ou $(E_{\text{rec}} \leq E_{\text{sup}})$;

L'étape terminale:

Si $E_{\text{rec}} < E_{\text{sup}}$ alors l'ensemble $E_{\text{sup}} = E_{\text{rec}}$.

Fin.

3.2. Nouvelle représentation de la liste fermée

L'algorithme recherche la plus petite surface qui contient toutes les pièces avec leurs copies. Il est basé sur deux listes principales ξ_2 fermée et ξ_1 ouverte. Ces dernières occupent une place mémoire très importante au cours de l'exécution. La nouvelle représentation de la liste fermée [2] permet de réorganiser la liste fermée d'une façon intelligente pour éviter les calculs inutiles.

Pour se faire, on fixe L_H et W_V d'après la solution initiale réalisable de Algo1[8] telles que :

L_H : La longueur du rectangle solution, obtenue par Algo1[8] en utilisant la construction horizontale,

W_V : La largeur du rectangle solution, obtenue par Algo1[8] en utilisant la construction verticale.

A chaque itération, on sélectionne R de ξ_1 , ayant la plus petite borne inférieure f^* et on le place dans ξ_2 . La liste ξ_3 contient les r-assemblage obtenus par les combinaisons horizontales de R avec tout R' de l'ensemble \mathcal{G}_{l_R} et les combinaisons verticales de R avec tout R'' de l'ensemble \mathcal{G}_{w_R} où \mathcal{G}_{l_R} et \mathcal{G}_{w_R} sont des sous-ensembles de ξ_2 tel que :

$$\begin{cases} \mathcal{G}_{l_R} = \{q / l_q = l_R + l_p \leq L_H, p \in \xi_2\} \\ \mathcal{G}_{w_R} = \{q / w_q = w_R + w_p \leq W_V, p \in \xi_2\} \\ \text{où } l_p \text{ et } w_p \text{ sont la longueur et la largeur de l'élément } p. \end{cases}$$

L'algorithme s'arrête dès que $f^*(R)$ de R sélectionné de ξ_2 est supérieure ou égale à la meilleure valeur de la solution courante. Sinon, ξ_1 est réduite à l'ensemble vide.

3.3. Notion de modèle dominé et d'ordre lexicographique

3.3.1. Notion de modèle dominé

La notion de modèle dominé [9] est adaptée à BFBB pour stériliser des r-assemblage comme suit :

Soient R et R' deux r-assemblages. La construction orthogonale entre R et R' est dite modèle dominé s'il existe $R'' \in \xi_1$ qui occupe l'espace vide S_{rem} obtenu par la construction orthogonale entre R et R' .

Cette technique est introduite dans BFBB de façon qu'à chaque nouveau modèle produit par combinaison orthogonale, on calcule l'espace vide. S'il existe une pièce de ξ_1 qui peut occuper cet espace sans dépasser les contraintes exigées, alors ce modèle est un modèle dominé à éliminer

3.3.2. Ordre lexicographique

L'ordre lexicographique [9] est appliqué à BFBB, afin d'éliminer les r-assemblages dupliqués modèles de symétrie. Il est introduit dans ξ_1 initiale, tel qu'à chaque R_i ($i \in I$), on associe deux ordres lexicographiques, un horizontal et un vertical, où $i = \theta_h = \theta_v = 1, 2, \dots, n$.

Pour chaque r-assemblage A, obtenu par construction horizontale entre K et Q, on introduit un nouvel ordre lexicographique tel que $\theta_{h(A)} = \min\{\theta_{h(K)}, \theta_{h(Q)}\}$ et $\theta = \theta_{v(A)} = \max_{E \in \xi_1 \cup \xi_2} (\theta_{v(E)} + 1)$

Test du r-assemblage dupliqué. Soient R et R' deux r-assemblages, $\theta_{h(R)}$ et $\theta_{h(R')}$ les ordres lexicographiques horizontaux de R et R' respectivement et $\theta_{v(R)}$ et $\theta_{v(R')}$ les ordres lexicographiques verticaux de R et R' respectivement.

Si $R \in$ liste ouverte composée d'au moins deux pièces et $R' \in$ liste fermée composée de pièces du même type, alors :

- La construction horizontale entre R et R' est un modèle dupliqué si : $\theta_{h(R)} < \theta_{h(R')}$
- La construction verticale entre R et R' est un modèle dupliqué si : $\theta_{v(R)} < \theta_{v(R')}$

Algorithme exact BFBB amélioré (BFBBA) :

ξ_1 : Ensemble des sous problèmes ;

ξ_2 : Liste de mémorisation des meilleurs sous problèmes ;

R, R' et R'' : r-assemblages ;

$f'(R)$: Borne inférieure du sous problème contenant R ;

Opt : Valeur de la meilleure solution actuelle ;

S_{rem} : Espace vide obtenu par la construction horizontale ou verticale entre R et R' ;

$\theta_{h(R)}$: Ordre lexicographique horizontal de R ;

$\theta_{v(R)}$: Ordre lexicographique vertical de R .

Entrée: Une instance du problème d'assemblage rectangulaire orthogonal

Sortie: La valeur de la solution optimale Opt

L'étape initiale:

$\xi_1 = \{R_1, \dots, R_n\}$; $\xi_2 = \emptyset$ et fin = faux ;

Soit E_{sup} la borne supérieure obtenue par l'application de Algo BS ; Opt = E_{sup} ;

L'étape principale:

Répéter

Choisir le r-assemblage R avec la plus petite valeur de f' ; (notée f'_{\min})

Si $\text{Opt} - f'_{\min} \leq 0$; alors fin = vrai

Sinon

Début

1. Transférer R de ξ_1 à ξ_2

2. construire ξ_3 tel que :

- a. Chaque R'' de ξ_3 obtenu par la construction horizontale entre R et les éléments \mathcal{G}_{l_R} de ξ_2 tel que $\mathcal{G}_{l_R} = \{q/l_R + l_p \leq L_H, p \in \xi_2\}$
- b. Chaque R'' de ξ_3 obtenu par la construction verticale entre R et les éléments \mathcal{G}_{w_R} de ξ_2 tel que $\mathcal{G}_{w_R} = \{q/w_q = w_R + w_p \leq W_V, p \in \xi_2\}$
- c. Tester R avec les éléments K de ξ_2 . Si le modèle obtenu est un modèle dupliqué, alors le nouveau modèle n'est pas construit.
- d. Chaque élément de ξ_3 satisfaisant les contraintes b_i ($1 \leq i \leq n$) et $f' < \text{Opt}$;
- e. Chaque modèle de ξ_3 qui satisfait le test de dominance est éliminé.
- f. Chaque élément de ξ_3 est étiqueté par un ordre lexicographique.

3. S'il \exists un t-assemblage $R' \in \xi_3 \setminus \{f'(R') < \text{Opt}\}$, alors $\text{Opt} = f'(R')$; mettre à jour ξ_1 par $\xi_1 \cup \xi_3$ et remplacer ξ_1 par $\xi_1 / \{\text{assemblage non terminal avec l'évaluation } f' \geq \text{Opt}\}$;

4. Si $\xi_1 = \emptyset$ alors fin = vrai ;

Fin de si

Jusqu'à fin = vrai ;

Fin.

4. Implémentation & Résultats

l'algorithme proposé a été codé en C et testé sur un ordinateur avec processeur Pentium 4 CPU 3.00 GHz, et 1 GO de RAM. La performance de notre algorithme (ABFBB) est évaluée sur plusieurs instances générées aléatoirement. Nous considérons un groupe de 50 instances, le nombre de pièces utilisées est pris dans l'intervalle [3, 16]. Les dimensions (l_i , w_i) de toutes les pièces sont fixées dans l'intervalle [1, 80], et la borne b_i $i = 1, \dots, n$, est générée aléatoirement dans l'intervalle [1, 9].

Performance de l'algorithme amélioré ABFBB

Table 1. Performances de l'algorithme BFBB amélioré

Borne	RM (Opt/BS)	RM (Opt/NBS)
Supérieure	1.21	1.15
BFBB	N. N. M 3638,58 N.N. M	T. M (s) 86.88 T. M (s)
BFBBA	2532,24	13.75

Table 2. Gain de l'algorithme BFBB amélioré

%Gain	N. N. M	T. M
(BFBBA/BFBB)	87,17	30,40

RM (Opt/BS) : Rapport Moyen entre Opt et Borne Supérieure (Algo1)[7] ;
 RM(Opt/NBS) : Rapport Moyen entre Opt et la Nouvelle Borne Supérieure (Algo BS) ;
 N.N.M : Nombre moyen de nœuds (construction) ;
 T.M(s) : Temps moyen d'exécution (en second).

Les résultats obtenus dans le tableau 1 indique que la nouvelle borne supérieure AlgoBS est meilleure que la borne supérieure Algo1[7], elle passe de 1,21 à 1,15 en moyenne. Les solutions obtenues de l' Algo BS permettent de réduire encore les constructions inutiles au niveau des nœuds internes.

Les résultats de la table 2 montrent clairement l'amélioration apportée par l'algorithme Best First Branch and Bound Amélioré (ABFBB). Le gain en moyenne sur le temps d'exécution est considérablement augmenté, il est égal à 84,17% et produit une réduction moyenne de 30,40% en terme de nombre de nœuds (construction) générés.

5. Conclusion

Nous avons développé une version améliorée de l'algorithme BFBB proposé par Hifi & Ouafi (1996) pour la résolution du problème d'assemblage orthogonal rectangulaire.

Dans la nouvelle version proposée, nous nous sommes basé sur trois techniques qui consiste en :

1. Développement d'une nouvelle borne supérieure (heuristique gloutonne)
2. Adaptation de la nouvelle représentation de la liste fermée (structure de données) afin d'accélérer la construction de la nouvelle configuration (VD Cung et al, 2000)
3. Introduction des stratégies des modèles dominés et modèles symétriques afin d'éliminer certains modèles et de restreindre la recherche de l'arbre développé.

Les résultats expérimentaux obtenus montrent que l'algorithme ABFBB est meilleur que l'algorithme BFBB en termes de temps d'exécution et nombre total de nœuds explorés.

Références

1. F. Clautiaux, A. Jouglet, J. Carlier, A. Moukrim (2006b), A new constraint programming approach for the orthogonal packing problem, *Computers and Operations Research*, doi:10.1016/j.cor.2006.05.012.
2. E. G. Coffman, and J. C. Lagarias (1989). Algorithms for packing squares: a probabilistic analysis. *SIAM Journal of Computing* 18 (1), 166-185.
3. V.D. Cung, M. Hifi and B.L. Cun (2000): Constrained two-dimensional cutting stock problems a best-first branch-and-bound algorithm. *International Transactions in Operational Research*.7 pp 185-210
4. Y. Cui, Y. Yang, X. Cheng, and P. Song,(2008) "A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem," *Computers & Operations Research*, vol. 35, no. 4, pp. 1281–1291.
5. A. J. W. Duijvestijn (1978). Simple perfect squared square of lowest order. *Journal of Combinatorial Theory* 25 (B), 260-263.
6. H. Dyckhoff (1990): A typology of cutting and packing problems. *European Journal of Operational Research* 44 145-159.
7. P. Gilmore, and R. Gomory (1965):Multistage cutting problems of two and more dimensions. *Ops Res.* 13, 94-119
8. M. Hifi (1994). Study of some combinatorial optimization problems: cutting stock, packing and Set covering problems, Ph. D. Thesis, Université de Paris 1 Panthéon-Sorbonne (in French).
9. M. Hifi and R. Ouafi (1998): A best-first branch and bound algorithm for orthogonal rectangular packing problem. *International Transactions in Operational Research* Vol.5, No.5, pp. 345-356
10. M. Hifi and R. M'hallah (2004): An exact algorithm for constrained two dimensional two-staged cutting problems. *Operations Research* Vol.53, No.1, pp 140-150

11. A.I. Hinxma. (1980). The tri-loss and assortment problems: a survey. *European Journal of Operational Research* 5 8-18.
12. S. Jakobs (1996). On genetic algorithm for the packing of polygons. *European Journal of Operational Research* 88, 165-181.
13. L. K. Kantorovich (1960): *Mathematical methods of organizing and planning production*. *Management Science* 6,363-422.
14. D.J. Kleitman and M.K. Krieger (1975): An optimal bound for two dimensional bin packing. *Proceedings of the 16 th Annual Symposium on foundations of computer Science*, pp. 163-168.
15. P. Sweeney, E. Paternoster (1992). *Cutting and Packing problems: a categorized application-oriented research bibliography*. *Journal of Operation Research Society* 43(7), 691-706.
16. G. Wäscher, H. Haußner, H. Schumann, (2004): *An Improved Typology of Cutting andPacking Problems*, 1st ESICUP Meeting, Lutherstadt Wittenberg, Germany, 18-20 March 2004
17. D. Zhang, Y. Kang, A. Deng (2006), A new heuristic recursive algorithm for the strip rectangular packing Problem, *Computers and Operations Research* 33(8); 2209-2217