



Approche algorithmique pour une classe de problème de découpe guillotine

Abdelkader Khelladi, Rachid Ouafi

Département de Recherche Opérationnelle
Faculté de Mathématiques
Université des sciences et de la technologie d'Alger
B.P. 32, Bab Ezzouar, 16111 Alger
Algérie.
Rachid_ouafi@hotmail.com

|||||
Résumé Le problème de découpe guillotine est un problème NP-complet et possède de nombreuses applications dans les processus de fabrication de diverses industries (textile, acier, bois, verre, etc. ...). Il s'agit d'optimiser l'utilisation d'une entité disponible en taille limitée en y découpant des sous-entités prédéterminées conformément à la découpe guillotine. Nous présentons une revue des approches algorithmiques développées pour différentes variantes du problème, tel que le problème de découpe sans contraintes et le problème de découpe généralisé.

Mots clés : Problèmes de découpe, méthodes arborescentes, programmation dynamique
|||||





1. Notions générales sur les problèmes de découpe

Du point de vue historique, le problème de découpe a été posé pour la première fois par Kantorovich [18]. Gilmore et Gomory [11] l'ont ensuite généralisé aux problèmes à deux et à plusieurs dimensions.

Par la suite, ce problème a été utilisé par d'autres chercheurs ([14,15]) pour l'étude d'autres variétés du problème de découpe (papier, bois, cuir, verre, etc.).

1.1 Famille de problème de découpe

Les problèmes de découpe sont de nature diverse. Cette diversité est reliée au support ou matériel considéré, aux types d'objets (qu'on appellera pièces) à placer et aux contraintes imposées sur la façon de découper le(s) support(s).

Formellement, une instance du problème de découpe à deux dimensions consiste à découper un support rectangulaire appelé *rectangle initial* de dimensions (L,H) en petites pièces rectangulaires de dimensions (l, h_i) et chacune de profit c_i pour $i=1, \dots, n$. L'objectif étant de maximiser la somme des utilités des pièces produites sur le(s) support(s) à découper. De plus, on dit que le problème de découpe est *non pondéré* si le profit de chaque pièce est égal à sa surface et *pondéré* sinon.

Des contraintes additionnelles sont imposées sur la manière de découper le(s) support(s) disponible(s) conformément au matériel de découpe utilisé. Une découpe qui produit deux rectangles est appelé *découpe guillotine*, précisément une *découpe guillotine* sur une plaque rectangulaire s'effectue d'une seule tranche en allant d'un côté de la plaque rectangulaire à son opposé horizontalement ou verticalement.

Un ensemble de pièces rectangulaires définit un modèle de découpe si ces pièces peuvent être obtenues par une séquence de découpes possibles sur le support disponible. Toutes les pièces produites qui diffèrent des pièces à découper sont considérées comme des chutes.

- ◆ Une possibilité conception de modèles de découpe passe par la génération de modèles de bande dont nous distinguons les catégories suivantes :
- ◆ Une bande générale horizontale (resp. verticale) est une bande composée d'au moins deux pièces de hauteur (resp. longueur) différentes.
- ◆ Une bande uniforme horizontale (resp. verticale) est une bande où toutes les pièces participantes sont de même hauteur (resp. longueur).
- ◆ Une bande homogène horizontale (resp. verticale) est une bande où il n'y a qu'un seul type de pièces participant.

Une bande optimale de longueur α et de hauteur h (resp. de hauteur β et de longueur l) : est une bande générale occupant la surface maximum de la bande (α, h) (resp. (l, β)).

Si on note par $S_{\alpha, h}(H^*)$, $S_{\alpha, h}(U^*)$, $S_{\alpha, h}(G^*)$ la surface maximale occupée dans une bande homogène, uniforme, et générale respectivement, de longueur α et de hauteur h , on a nécessairement $S_{\alpha, h}(H^*) \leq S_{\alpha, h}(U^*) \leq S_{\alpha, h}(G^*)$.



Un modèle de découpe réalisable s'obtient donc naturellement comme combinaison d'un certain nombre de bandes horizontales ou verticales. Nous retenons les différents types de modèles de découpes suivants :

1. Un modèle de découpe *général* (ou dissection générale) est un modèle constitué par la combinaison de bandes générales verticales et horizontales.
2. Un modèle de découpe *uniforme* est un plan de découpe réalisable constitué uniquement de bandes uniformes.
3. Un modèle de découpe *homogène* est une dissection uniforme pour lequel la solution est caractérisée par la répétition d'un seul type de pièces.

1.2 Hypothèses sur les différents problèmes

Dans notre étude, les algorithmes développés sont dédiés aux problèmes de découpe vérifiant les hypothèses suivantes :

- Les découpes effectuées sur le rectangle initial, sont du type guillotine.
- Les pièces à découper sont fixées, une pièce de longueur l et de hauteur h est différente de celle de longueur h et de hauteur l .
- Les dimensions des supports et des pièces sont des entiers.

2. Un nouvel algorithme pour la résolution du problème de découpe sans contraintes à deux dimensions

Le problème de découpe guillotine non contraint à deux dimensions qui est la forme standard du problème de découpe dans la littérature. Il s'agit d'optimiser l'utilisation d'une entité disponible en taille limitée en y découpant des sous-entités prédéterminées. L'entité disponible est sous forme rectangulaire de dimensions fixées, qu'on appelle rectangle initial. Les sous-entités ont aussi des formes rectangulaires qu'on appelle pièces disponibles. L'objectif étant de découper le rectangle initial en pièces appartenant à l'ensemble des pièces disponibles S tout en minimisant la surface des pièces produites n'appartenant pas à cet ensemble, appelé chute (version non pondérée du problème) ou en maximisant le profit total sur les ensembles des pièces à découper (version pondérée).

Nous présentons une nouvelle approche de résolution du problème de découpe sans contraintes applicable aux deux versions pondérée et non pondérée, notre méthode s'appuie sur des techniques de programmation dynamique, et une recherche arborescente avec critère de sélection du meilleur d'abord.

2.1 Présentation du problème

Une instance du problème de découpe sans contraintes qu'on note par TDGC est définie par le triplet (R, S, c) . $R = (L, H)$ est le rectangle initial, où L et H sont sa longueur et sa hauteur respectivement. Les sous-entités rectangulaires sont représentées par l'ensemble $S = \{p_1, \dots, p_n\}$ dans lequel chaque pièce p_i est de dimensions $(l_i, h_i) \in (L, H)$.

$c = (c_1, \dots, c_n)$ est le vecteur poids (la pièce i a un poids c_i). Le problème consiste à découper le rectangle disponible R en petites pièces rectangulaires appartenant à S tel que le profit

enregistré $F(x) = \sum_{i=1}^n c_i x_i$, soit de valeur maximale, où x_i désigne le nombre de pièces de type i produites sur le support à découper R .



On dit que le problème TDGC est *non pondéré* si le profit c_i associée à la pièce i est égale à sa surface ($c_i = l_i h_i$), sinon le problème est dit *pondéré*.

Par ailleurs, tout programme mathématique dédié au problème TDGC nécessite la prise en charge d'une part, des contraintes additionnelles qui garantissent que les dimensions L et H du support ne seraient pas débordées et d'autre part, que les plans de découpe réalisés soient conforme à la découpe guillotine.

2.2 Principe de l'algorithme

Notre approche notée BFS se base sur une procédure constructive permettant d'obtenir des modèles de découpe guillotine sur l'entité initiale R au moyen de constructions verticales et horizontales.

Définitions 1.

- Une construction verticale de deux (sous-) rectangles de dimensions (l_1, h_1) et (l_2, h_2) est représentée par le (sous-) rectangle de dimensions $(\max\{l_1, l_2\}, h_1 + h_2)$.
- Une construction horizontale produit un (sous-) rectangle de dimensions $(l_1 + l_2, \max\{h_1, h_2\})$.
- Dans les deux cas de figures *a)* et *b)*, chaque (sous-) rectangle R résultant de construction verticale ou de construction horizontale vérifiant $(l_R, h_R) \leq (L, H)$ est considérée comme un *rectangle guillotine*.

A chaque rectangle guillotine R construit, on fait correspondre une fonction d'utilité $g(R)$ et une fonction complémentaire $h(R)$.

$g(R)$: représente la somme des profits des pièces constituant R .

$h(R)$: désigne une estimation du plan de découpe sur le reste de la surface notée F (voir Figure 1). En fonction de ces paramètres,

$$f(R) = g(R) + h(R)$$

constitue la valeur maximale du modèle de découpe faisable contenant le rectangle guillotine (construit) R .

Ce processus de construction est schématisé par une arborescence dans laquelle chaque nœud désigne un niveau de construction.

Les évaluations des bornes inférieures et supérieures sont réalisées en modélisant le problème sous forme de problèmes de knapsacks unidimensionnels résolus par la programmation dynamique.



Figure 1 : les paramètres d'évaluation d'un rectangle guillotine



De plus à chaque niveau de l'arborescence, nous exploitons les sous problèmes induits (par la programmation dynamique) pour calculer les bornes inférieures et supérieures des autres sommets intérieurs. Par ailleurs ces bornes sont aussi utilisées pour la stérilisation de certaines branches par :

- la récupération d'une solution à partir d'un niveau inférieur de l'arborescence.
- réutilisation des résultats issus de la programmation dynamique des nœuds en cours d'exploration.

2.2.1 Bornes supérieures sur les sommets

Un plan de découpe optimale pour la région F ne dépasse en aucun cas la surface de celle-ci, les résultats suivants donne une description du traitement de la région complémentaire F en tant que surface (barre), ainsi une estimation d'une borne supérieure pour $h(R)$ peut être obtenue par la résolution d'un knapsack unidimensionnel relaxé.

Théorème 2.1 : Soit R un rectangle guillotine de dimensions (l_R, h_R) , une borne supérieure pour $f(R)$ est donnée par :

$$f(R) = g(R) + h'(R)$$

$$\text{où : } h'(R) = l_R h_R \cdot \max \left\{ \frac{c_j}{l_j h_j}, j \in F \right\}$$

est la valeur de la solution du problème de sac à dos unidimensionnel relaxé suivant :

$$K(F) \begin{cases} \text{Max } \sum_{j \in F} c_j x_j \\ \text{s.c. } \sum_{j \in F} (l_j h_j) x_j \leq S(F) \\ 0 \leq x_j \leq nb_j(F_1 \cup F_2), j \in F \end{cases}$$

$S(F)$ désigne la surface de F . F_1 and F_2 les sous-rectangles représentés respectivement par (L, l_R, H) et $(L, H - h_R)$. $nb_j(y)$ est le nombre maximum d'apparition de la pièce j dans le sous-rectangle y . \square

Corollaire 2.1 :

Une borne supérieure initiale pour le rectangle R est donnée par :

$$Upper(R) = L.H. \max \left\{ \frac{c_j}{l_j h_j}, j \in S \right\}$$

2.2.2 Borne inférieure initiale

La borne de départ de notre approche nécessite la résolution de quatre knapsacks unidimensionnels, deux knapsacks engendrent les différentes bandes optimales horizontales et verticales et deux autres knapsacks unidimensionnels permettent de construire deux modèles de découpe réalisables, le premier est un plan de découpe horizontal obtenu comme combinaison des bandes optimales de différentes hauteurs, le deuxième est un plan de découpe vertical

obtenu en combinant les bandes optimales verticales de différentes longueurs. La borne inférieure initiale est choisie comme la valeur du meilleur modèle de découpe parmi ces modèles construits.

La modélisation des knapsacks unidimensionnels repose sur les résultats suivants [17]

Génération des bandes optimales :

Nous ordonnons les éléments de S par ordre croissant tel que : $h_{k_1} \leq h_{k_2} \leq \dots \leq h_{k_r}$ où r est le nombre des différentes hauteurs des pièces de l'ensemble S , et $R = (L, H)$ désigne le rectangle initial.

Proposition 2.1. : Toutes les bandes optimales horizontales des différentes hauteurs et de longueur L sont générées par la résolution d'un seul problème de sac à dos.

$$K_{LH}^r \left\{ \begin{array}{l} \max F_r(L) = \max \sum_{j=1}^n c_j x_j \\ \text{s.c.} \sum_{j=1}^n l_j x_j \leq L \\ x_j \in \mathbb{N} \end{array} \right.$$

Si nous considérons maintenant tous les éléments de S ordonnés selon un ordre croissant sur les longueurs, tel que : $l_{k_1} \leq l_{k_2} \leq \dots \leq l_{k_{r'}}$ où r' est le nombre des différentes longueurs de l'ensemble des pièces S . Alors les différentes bandes verticales sont générées en vertu du résultat suivant :

Proposition 2.2. : Toutes les bandes optimales verticales des différentes longueurs et de hauteur H sont générées par la résolution d'un seul problème de sac à dos.

$$K_{HL}^{r'} \left\{ \begin{array}{l} \max F_{r'}(H) = \max \sum_{j=1}^n c_j x_j \\ \text{s.c.} \sum_{j=1}^n h_j x_j \leq H \\ x_j \in \mathbb{N} \end{array} \right.$$

Modèles de découpe :

La sélection des meilleures bandes conduit à la construction de modèles de découpe réalisable. Les deux résultats suivants nous montrent comment effectuer le choix du meilleur plan de découpe horizontal (respectivement vertical) généré à partir de combinaisons linéaires des éléments de l'ensemble des bandes horizontales (respectivement verticales).

Proposition 2.3. : Un plan de découpe horizontal pour le rectangle initial $R = (L, H)$ est obtenu par la résolution du problème de sac à dos suivant :

$$K_z^{hor}(H) \left\{ \begin{array}{l} B_{hor} = \max \sum_{k=1}^r F_k(L) y_k \\ \text{s.c.} \sum_{k=1}^r h_k y_k \leq H \\ y_k \in \mathbb{N} \end{array} \right.$$

où : r est la cardinalité de l'ensemble des bandes horizontales de longueur L , y_k est le nombre d'apparition de la $k^{ème}$ bande horizontale de hauteur h_k et de profit $F_k(L)$. \square

Proposition 2.4 : Un plan de découpe vertical pour le rectangle initial $R=(L,H)$ est obtenu par la résolution du problème de sac à dos suivant :

$$K_y^{ver}(L) \left\{ \begin{array}{l} B_{ver} = \max \sum_{k=1}^{r'} F_k(H) y_k \\ \text{s.c.} \sum_{k=1}^{r'} l_k y_k \leq L \\ y_k \in \mathbb{N} \end{array} \right.$$

où : r' est la cardinalité de l'ensemble des bandes verticales de hauteur H , y_k est le nombre d'apparition de la $k^{ème}$ bande verticale de longueur l_k et de profit $F_k(H)$. \square

La borne inférieure initiale est donnée par $B_{inf}(R) = \max(B_{hor}, B_{ver})$

2.2.3. Evaluation des bornes inférieures sur les sommets intérieurs

Pour tout rectangle guillotine R , on peut lui associer deux solutions faisables possibles en subdivisant la région F en deux rectangles : un plus petit noté F_1 voir figure 2a. (ou F_2 sur la figure 2b) et un plus grand F_2 figure 2a. (ou F_1 sur la figure 2b). Pour ces deux rectangles, on effectue les opérations suivantes :

1. On détermine le plan de découpe associé au plus grand rectangle, cette valeur est déjà disponible à partir des procédures de résolution utilisant la programmation dynamique des knapsacks K_z^{hor}, K_y^{ver} . La valeur de la solution obtenue est notée L_1 .
2. On complète la solution faisable avec la meilleure solution homogène sur le petit rectangle, on note la valeur de la solution par L_2 .

Finalement, la valeur du meilleur modèle de découpe faisable associé au sous-rectangle construit R est donnée par $Lower(R) = g(R) + L_1 + L_2$

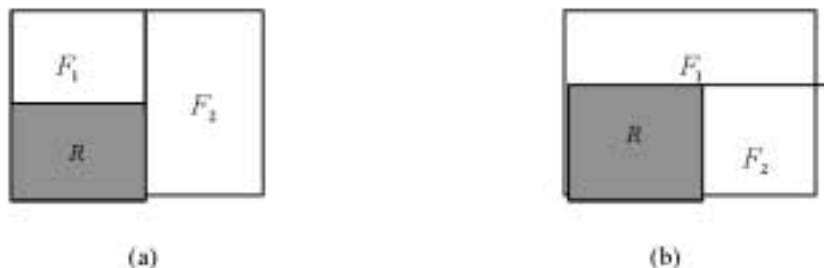


Figure 2. Représentation de deux modèles de découpe faisables (a) et (b) pour l'évaluation d'une borne inférieure sur le rectangle R.

Le résultat qui suit résume de manière explicite les procédures utilisées pour l'évaluation des bornes inférieures sur les sommets intérieurs. Nous soulignerons à cet effet, que les valeurs nécessaires pour l'évaluation des rectangles les plus grand (de la région complémentaire F) pour les deux solutions alternatives envisagées, sont déjà disponibles à partir de la procédure de résolution par la programmation dynamique des knapsacks unidimensionnels K_L^{hor} et K_H^{ver} utilisés pour l'évaluation de la borne inférieure initiale

Théorème 2.2 :

Etant donné un rectangle guillotine R de dimension (l_n, h_n) , F_1 et F_2 deux rectangles de dimensions $(L, H - h_n)$ et $(L - l_n, H)$, respectivement. Une borne inférieure pour $f(R)$ est :

$$lower(R) = \begin{cases} g(R) + B_L^{hor}(H - h_n) + H^*(F_2) & \text{si } S(F_1) \geq S(F_2) \\ g(R) + B_H^{ver}(L - l_n) + H^*(F_1) & \text{si } S(F_1) < S(F_2) \end{cases}$$

où : $B_L^{hor}(H - h_n)$ est la valeur de la solution du knapsack $K_L^{hor}(H - h_n)$, $B_H^{ver}(L - l_n)$ est la valeur de la solution du knapsack $K_H^{ver}(L - l_n)$, $H^*(F_1)$ et $H^*(F_2)$ étant la valeur du meilleur plan de découpe homogène pour F_1 respectivement F_2 .

3. Approche séquentielle exacte pour la résolution du problème de découpe généralisé sans contraintes

Le problème de découpe avec un seul support en stock est en réalité un cas particulier du problème appelé le problème de découpe généralisé à deux dimensions (GTDGC). Nous proposons un algorithme exact pour la résolution du problème de découpe généralisé sans contraintes, le principe de la méthode développée repose sur l'application de la programmation dynamique, et en particulier le principe de Gomory [11], et l'introduction d'une nouvelle stratégie basée sur le principe de saut de points, qui permet de réduire l'espace de recherche.

3.1 Présentation du problème

Une instance du problème de découpe généralisé sans contraintes noté GTDGC est représenté par m supports rectangulaires R_j de dimensions (L_j, H_j) , $j = 1, \dots, m$, un ensemble de n pièces



rectangulaires $S=\{(l_1,h_1),\dots,(l_m,h_m)\}$, et un vecteur de profit c avec c_i le profit de la i^{me} pièce. L'objectif étant de découper l'ensemble des pièces sur les supports en stock de façon à maximiser la rentabilité de chaque support, notée : $F_j(x_j) = \sum_{i=1}^m c_i x_{(i,j)}$, où $x_{(i,j)} \in \mathbb{N}$ désigne le nombre de pièces de type i produites sur le support j , pour $j = 1,\dots,m$.

Nous supposons que les pièces sont fixées, que la contrainte de découpe est du type guillotine, et le nombre de découpe guillotine est illimité (cas sans contraintes de niveaux).

3.2 Principe de l'approche exacte

Définition 3.1 : Un support rectangulaire non dominant est un support rectangulaire dont les dimensions $(L_k,H_k)_{k \leq m}$ vérifient l'inégalité suivante :

$$\exists k' \neq k \wedge l \leq k' \leq m \setminus L_k \leq L_{k'} \wedge H_k \leq H_{k'}$$

Notre approche utilise une procédure qui effectue au préalable, un tri sur l'ensemble des supports en stock, en vue de recenser les éventuels supports *non-dominants*. Nous démontrons que la présence de tels supports peut être omise sans affecter la solution optimale du problème.

Par la suite, nous réordonnons les supports restants selon un ordre croissant sur les longueurs. Nous appliquons une procédure basée sur la programmation dynamique sur des rectangles fictifs en utilisant le principe de saut de points qui consiste à :

- Détecter à l'intérieur du rectangle (fictif) considéré les parties communes des supports qui peuvent être traités simultanément.
- Négliger les parties non concernées.

La figure 3 illustre l'application de la procédure dans le traitement de trois supports en stock.

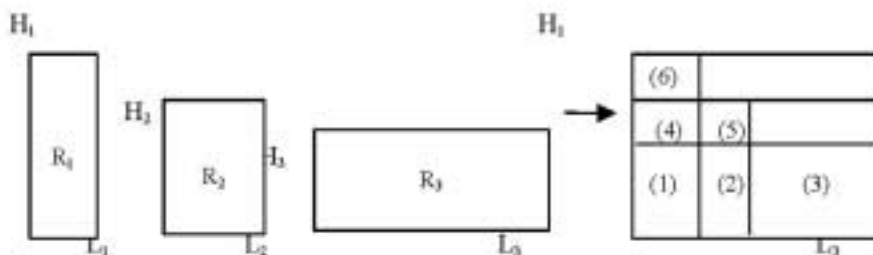


Figure 3

(a) Trois rectangles distincts en stock,

(b) Traitement des parties communes avec le principe de saut de points.



Les parties non numérotées représentent l'ensemble des points non traités par l'algorithme.

Nous démontrons que si toutes les longueurs des supports respectaient un ordre croissant, alors nécessairement toutes les hauteurs respectaient l'ordre inverse. Finalement, en utilisant l'ordre sur les longueurs, nous démontrons la convergence de l'algorithme.

Lemme 3.1 : Soit k un indice d'un support rectangulaire. S'il n'existe pas de supports rectangulaires non dominants dans la liste des supports en stock, alors

$$\forall k \mid 1 \leq k \leq m-1 : L_k \leq L_{k+1} \wedge H_k \geq H_{k+1}$$

Théorème 3.1 : L'algorithme EAG détermine les valeurs de la solution optimale pour tous les supports rectangulaires en stock (L_j, H_j) , pour $j = 1, \dots, m$ □

5. Conclusion générale et perspective de recherche

Les approches que nous avons été amenés à développer pour la résolution d'une classe difficile de problèmes de l'optimisation combinatoire, en l'occurrence les problèmes de découpe, utilise une multitude de variétés de techniques performantes issues aussi bien de la recherche opérationnelle (programmation dynamique, méthodes heuristiques, etc.) que d'autres branches telle que les méthodes d'intelligence artificielle (algorithme A*, stratégies de Hill climbing).

L'objectif principal consistait à réduire l'effort de recherche dans l'exploration des arborescences. Pour ce faire, nous avons doté les approches élaborées de règles de cheminement et de procédures d'évaluation efficaces en affinité profonde avec le schéma itératif global, capables d'exploiter au maximum l'information résultant de chaque calcul en un sommet visité.

En procédant de la sorte, il est apparu que l'on pouvait réduire de manière considérable le nombre de sommets à parcourir. De plus, la qualité des solutions obtenues est très proche de l'optimum pour les méthodes approchées.

Nous estimons qu'il serait sans doute fructueux d'élargir les champs d'application de ces genres d'approches intelligentes à d'autres variantes plus complexes, que nous souhaiterons traiter en perspective, comme les problèmes de découpe non guillotine et les problèmes de découpe à trois dimensions.

6. Bibliographie :

- [1] M. Adamowicz and A. Albano, *Two-stage solution of the cutting stock problem*, Information Procc. North Holland, vol. 71, p.p. 1086-1091, 1972.
- [2] R. Andonov, F. Rimbault and P. Quinton, *Dynamic programming parallel implementations for the knapsack problem*, Preprint IRISA, No 740, July 1993, Campus de Beaulieu, 35042 Rennes cedex .
- [3] M. Arenales and R. Morabito, *An and/or- graph approach to the solution of two dimensional non guillotine cutting problems*, European Journal of Operational Research, vol. 84, p.p. 599-617, 1995.
- [4] J. E. Beasley, *Algorithms for unconstrained two-dimensional guillotine cutting*, Journal of the Operational Research Society, vol. 36, p.p. 297-306, 1985.
- [5] N. Christofides and C. Whitlock, *An algorithm for two-dimensional cutting problems*, Operations Research, vol. 2, p.p. 31-44, 1977.

- [6] J. Daniels and P. Ghandforoush, *An improved algorithm for the non-guillotine constrained cutting-stock problem*, Journal of operation research society, vol. 41, N°2, p.p. 141-150, 1990.
- [7] K. Dowsland and W. Dowsland, *Solution approaches to irregular nesting problems*, European Journal of Operational Research, vol. 84, p.p. 506-521, 1995.
- [8] R.G. Dyson and A.S. Gregory, *The cutting stock problem in the flat glass industry*, Operation Research Quart., vol. 25, p.p. 41-53, 1974.
- [9] M.R. Garey and D.S. Johnson, *Computers and intractability. A guide to the theory of NP-completeness*, W.H. Freeman and Company, San Francisco, 1979.
- [10] P. Gilmore and R. Gomory, *Multistage cutting problems of two and more dimensions*, Operations Research, vol. 13, p.p. 94-119, 1965.
- [11] P. Gilmore and R. Gomory, *The theory and computation of knapsack functions*, Operations Research, vol. 14, p.p. 1045-1074, 1966.
- [12] F. Glover, and H. J. Greenberg, *New approaches for heuristic search: A bilateral linkage with artificial intelligence*, European Journal of Operational Research, vol. 39, p.p. 119-130, 1989.
- [13] E. Hadjiconstantinou and N. Christofides, *An optimal algorithm for general orthogonal 2-D cutting problems*, Technical Reports MS-91/2, Imperial College, London, 1991.
- [14] S.G. Hahn, *On the optimal cutting of defective sheets*, Operations Research, vol. 16, p.p.1100-1114, 1968.
- [15] S.S. Heragu, and A. Kusiak, *Machine layout problem in flexible manufacturing systems*, Operations Research, vol. 36, No 2, p.p. 258-268, 1988.
- [16] J. C. Herz, *A recursive computing procedure for two-dimensional stock cutting*, IBM Journal of Research and Development, vol. 16, p.p. 462-469, 1972 .
- [17] M. Hifi, *Exact algorithms for the guillotine strip/packing problem*, Computers and Operations Research, vol.25, N°11, p.p. 925-940, 1998.
- [18] L. K. Kantorovich, *Mathematical methods of organizing and planning production*, Management Science, vol. 6, p.p. 363-422, 1960.
- [19] R. Morabito, M. Arenales and V. Arcaro, *An and/or-graph approach for two-dimensional cutting problems*, European Journal of Operational Research, vol. 58, No 2, p.p. 263-271, 1992.
- [20] R. Ouafi and M. Hifi, *A best first branch-and-bound algorithm for orthogonal rectangular packing problems*, International Transactions in Operational Research, vol. 5, N°5, p.p. 345-356, 1998.
- [21] P.Y. Wang, *Two algorithms for constrained two-dimensional cutting stock problem*, Operations Research. Vol 31, p.p. 573-586, 1983.