

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTER DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOD MAMMERI DE TIZI-OUZOU



Faculté sciences /Département mathématiques

Mémoire de fin d'études

En vu d'obtention du diplôme de master en :
Mathématiques appliquées à la gestion d'entreprises

Intitulé :

Problème de placement , application et résolution

Réalisé par :

Saad Louiza
Leham Aldjia

Dirigé par :

Mr.KASDI

Année universitaire 2018/2019

Remerciement

On remercie dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

*Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de **Mr K.KASDI**, on le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.*

Je remercie mes très chers parents, qui ont toujours été là pour moi. Je remercie ma sœur mes frères pour leurs encouragements et Rien n'est possible sans vous.

Enfin, je remercie mes amis qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Dédicaces

*A la lumière de mes jours, la source de mes efforts, la
flamme de mon cœur, ma vie et mon bonheur; maman que
j'adore.*

*Aux personnes dont j'ai bien aimé la présence dans ce jour,
mes frères LYES ,MOUNIR et mes sœur ASSIA, LYLIA
SIHEM ,MELISSA ,ZAHIA .*

A toute ma famille, et mes amis,

*A mon binôme **LOUIZA** et toute la famille **SAAD**.*

*Et à tous ceux qui ont contribué de près ou de loin pour que ce
projet soit possible, je vous dis merci.*

*Aux personnes qui m'ont toujours aidé et encouragé, qui
étaient toujours à mes côtés, et qui m'ont accompagnaient
durant mon chemin d'études supérieures, mes aimables amis,
collègues d'étude, et sœurs de cœur , KHALOUDJA,
HOURIA ,DIHIA,KARIMA,NASSIMA,LYNDA.*

ALDJIA

Dédicaces

Al'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, à toi mon père.

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur; maman que j'adore.

Aux personnes dont j'ai bien aimé la présence dans ce jour, mon fiancé FARID, et bien sur à tous mes frères SOFIANE, YUCEF et NACER et ma sœur SONIA, mes nièces MASTENE et MEKAYELA, à toute la famille ZEMOUCHE et AMRANE, sans oublier mes beaux-parents que j'aime.

A toute ma famille, et mes amis,

*A mon binôme **ALDJIA** et toute la famille **LEHAM**.*

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagné durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, et frères de cœur, BRAHIM, ABDOU et DJAFAR mes copines de cœurs

LYLIA, ASSIA, DEHBIA et KHELOUDJA

LOUIZA

Sommaire

Introduction générale	1
Chapitre I Optimisation Combinatoire	
1. Introduction	4
2. Problème d'optimisation combinatoire	5
3. Eléments de la théorie de la complexité	14
4. Exemples de problème d'optimisation combinatoire	17
5. Conclusion	18
Chapitre II Le Problème de Bin Packing	
1. Introduction	19
2. Domaines d'application du problème de bin packing	19
3. Contraintes pratiques et classification	20
4. Le problème de bin packing uni-, bi- et tri-dimensionnel	21
Chapitre III Le placement dans l'industrie	
1. Introduction	28
2. Notions générales sur les problèmes de placement	28
3. Types de problèmes de placement	31
3. Propriétés matérielles	32
4. Processus de placement	34
5. Planification	35
6. Processus de coupe	35
7. Les différents types de découpe	36
8. Industries avec des problèmes de placement rectangulaires	38
9. Conclusion	41

Chapitre IV Résolution des problèmes de placement avec des pièces Rectangulaires

Introduction.....42

Partie I L'heuristique Least Wasted First (LWF)

1. Introduction.....43

2. Définition du problème.....43

3. Résolution du problème.....45

4. Présentation graphique.....52

Partie 2 Logiciel OptiCoup V

1. Introduction.....58

2. Principes généraux.....59

3. Saisie des données61

4. Edition des résultats66

5. Import et Export.....68

Chapitre V Application pratique

1. Introduction.....71

2. Application71

3. Résolution par l'heuristique LWF73

4. Résolution par OptiCoup V85

Conclusion générale90

Bibliographie.....91

Introduction générale

Introduction générale

La recherche opérationnelle est la discipline des mathématiques appliquées liée à l'informatique, qui traite des questions d'utilisation optimale des ressources dans l'industrie et dans le secteur public.

Elle est définie comme l'ensemble des méthodes et techniques rationnelles d'analyse et de synthèse des phénomènes d'organisation utilisables pour élaborer **de meilleures décisions**, tout en proposant des modèles conceptuels permettant d'analyser et de maîtriser des situations complexes pour permettre aux décideurs de comprendre et d'évaluer les enjeux afin de faire les choix les plus efficaces.

L'application de la Recherche Opérationnelle s'est élargie dans cette dernière décennie à divers domaines comme l'économie, la finance, le marketing et la planification d'entreprise. Plus récemment, elle a été utilisée pour la gestion des systèmes de santé et d'éducation, pour la résolution de problèmes environnementaux et dans d'autres domaines d'intérêt public.

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Les problèmes d'optimisation sont utilisés pour modéliser de nombreux problèmes dans différents secteurs de l'industrie (télécommunications, électronique, mécanique, chimie, transport, ...).

Dans le secteur industriel, la matière première est une composante très importante lors du calcul des coûts de production. Il est donc prioritaire de minimiser les pertes et d'améliorer sans cesse son utilisation dans l'intérêt de réduire les coûts, et pour aller plus loin réduire les impacts environnementaux engendrés par l'accumulation des déchets.

Le problème de placement, découpe et conditionnement, des variantes d'un problème d'intérêt majeur qui est le problème de **Bin Packing**. Dans l'industrie il est l'un des problèmes d'optimisation combinatoire, où il est difficile d'assurer une solution optimale dans les cas complexes. Il s'est avéré que l'utilisation des méthodes exactes est pratiquement très difficile à réaliser puisqu'elles nécessitent un temps d'exécution insupportable, l'utilisation des heuristiques approximatives s'avère donc très intéressante.

Ce problème consiste principalement à placer des articles en utilisant de la manière la plus économique possible des matériaux sous forme de feuille ou de rouleaux. Les contraintes liées aux objets varient selon la dimension du problème. Il comporte trois versions selon la dimension : la première en une dimension (1D), la deuxième en deux dimensions (2D) et la troisième en trois dimensions (3D).

Ces problèmes appartiennent à la catégorie des problèmes NP-difficile dans la classification de la complexité des problèmes.

Objectif

Modélisation et résolution d'un problème de placement qui concerne des pièces de formes rectangulaire Par une heuristique récente bien défini qui s'adapte à ce type de problème et de le résoudre avec un logiciel puis de comparer les résultats obtenues

Organisation du mémoire

Afin de bien présenter notre travail nous l'avons structuré de la manière suivante

➤ **Chapitre 1 : l'optimisation combinatoire**

La première partie de notre travail est consacrée à l'expose des techniques d'optimisation capables de résoudre un certain nombre de problèmes. Deux grandes classes de méthodes sont présentées : les méthodes exactes qui consistent généralement à énumérer, de manière implicite, l'ensemble des solutions de l'espace de recherche et garantissent de trouver une solution optimale, et les méthodes approchées qui traitent généralement des problèmes de grande taille, elles n'assurent pas de trouver la solution optimale mais sont efficaces. Les méthodes les plus connues et les plus utilisées vont être détaillées dans notre travail.

➤ **Chapitre 2 : présentation de problème de bin packing**

Ce chapitre décrit concrètement le problème de bin packing, les contraintes liées au problème, ses variantes, ainsi que quelques-unes de ses applications.

➤ **Chapitre 3 : Le placement dans l'industrie**

Un aperçu sur le monde de placement est donné. On décrira les types de problèmes les propriétés des matériaux et les différents type de découpe puis on terminera par présenter les problèmes de placement rectangulaires.

➤ **Chapitre 4 : Résolution des Problèmes de placement avec des pièces rectangulaires**

Dans ce chapitre, nous définissons le problème de placement avec les pièces rectangulaires, nous avons développé l'heuristique Least Wasted First(LWF) pour résoudre ce problème en précisant son fonctionnement ainsi que les étapes de son déroulement, cette heuristique récemment développé donne à priori de bons résultats.

Nous avons aussi résolu ce problème en utilisant le Logiciel OptiCoup V très utilisé dans le marché, On a expliqué son mode de fonctionnement détaillé afin de l'utiliser en pratique.

➤ **Chapitre 5 : Application pratique**

Afin de concrétiser les méthodes présentées dans le chapitre 4 nous avons traité une application pratique.

Nous présentons l'application puis les résultats obtenus en utilisant l'heuristique LWF et le logiciel OptiCoup V

Enfin, nous achevons ce mémoire par une conclusion générale qui exposera les perspectives envisagées.

Chapitre I

Optimisation Combinatoire

1. Introduction

L'optimisation combinatoire appelée aussi optimisation discrète, est une branche de l'optimisation en mathématiques appliquées et en informatique, elle est également liée à la recherche opérationnelle, à l'algorithmique et à la théorie de la complexité. Une des raisons de son développement est liée au nombre considérable de problèmes qu'elle permet de traiter, aussi bien dans les domaines de l'ingénieur que dans les domaines de recherche, ce qui a fait surgir un large éventail de concepts qui continuent d'être enrichies par une bibliographie très abondante et très dense, qui traite divers problèmes d'optimisation ainsi que leurs méthodes de résolution.

La résolution d'un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret de solutions appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais de cardinalité très grande [1].

Généralement, il s'agit pour les problèmes d'optimisation d'identifier un objectif pour mesurer la qualité d'un choix donné (gain financier, coût, temps, consommation de ressource ... etc.) en déterminant les éléments agissant sur cet objectif (variable de décision) et la manière dont ils interagissent entre eux (contraintes de système), ce processus est appelé modélisation.

Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir et à modéliser, leurs résolutions est généralement difficile, et la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficile et ne possèdent donc pas à ce jour des solutions algorithmiques efficaces pour toutes les données. [2]

Dans ce chapitre, nous présentons d'abord les éléments de base de cette discipline et sa relation avec la théorie de la complexité, en suite nous donnerons une classification simple des différentes classes de problèmes d'optimisation, et nous terminerons par quelques exemples et certaines méthodes de résolution.

2. Problèmes d'optimisation combinatoire

2.1 Definition

L'optimisation combinatoire consiste à minimiser ou maximiser une fonction souvent appelée fonction coût, d'une ou plusieurs variables soumises à des contraintes. L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire [3].

D'un point de vue mathématique, un problème d'optimisation combinatoire est une minimisation ou bien une maximisation d'une certaine fonction sous des contraintes. Notons par :

- x : le vecteur constitué de variables de décision, avec $x=(x_1,x_2,\dots,x_n)\in\mathbb{R}^n$,
- f : la fonction objectif réelle,
- g : le vecteur des fonctions intervenant dans les contraintes.

Un problème d'optimisation peut alors être défini en général de manière suivante :

$$\text{Minimiser } f(x) \quad (1.1)$$

$$\text{s.c} \quad \begin{aligned} g_i(x) &= 0, \quad i \in I_1 \\ g_i(x) &\leq 0, \quad i \in I_2 \end{aligned}$$

Où I_1 et I_2 sont les ensembles d'indices des contraintes de types égalité et inégalité respectivement.

2.2 Classification des problèmes d'optimisation

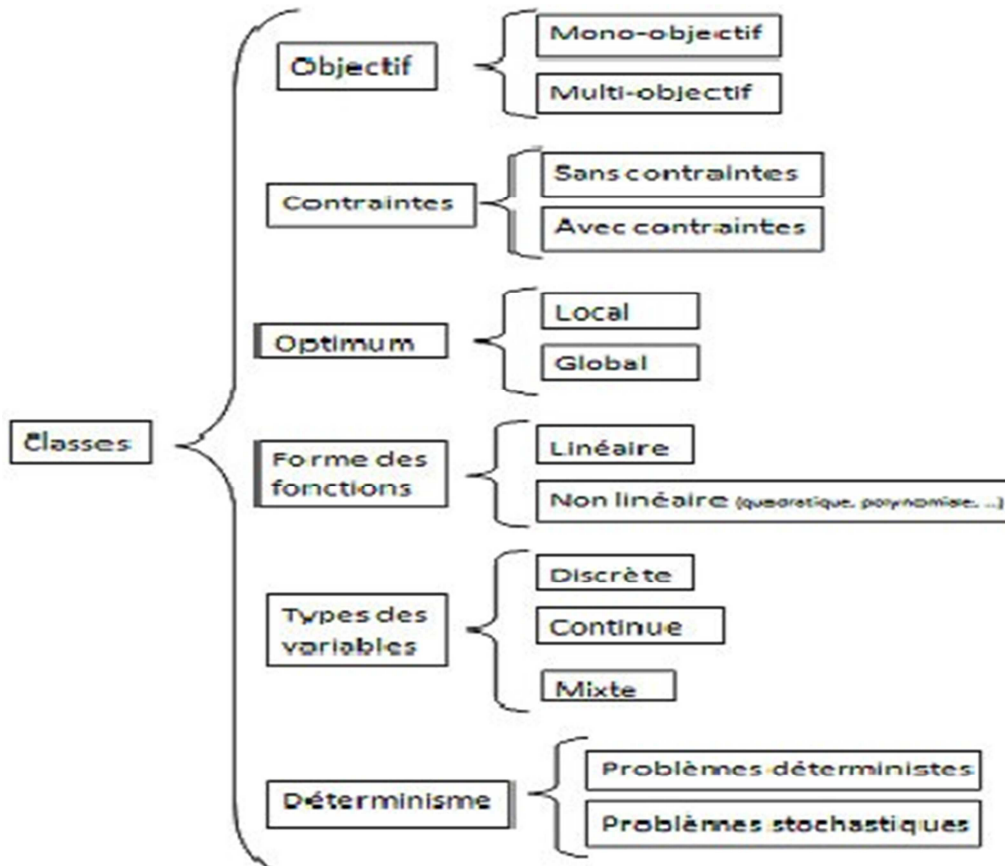


Figure 1 : Classification des problèmes d'optimisation

Face à la résolution d'un problème d'optimisation, il est important de bien identifier à quelle catégorie ce problème appartient. En effet les algorithmes développés sont conçus pour résoudre un type de problème donné et sont peu efficaces pour un types différent. La classification des problèmes d'optimisation change d'un auteur à l'autre. Par exemple, on distingue :

➤ **Les problèmes d'optimisation mono-objectif ou multi-objectif**

Les problèmes mono-objectif sont définis par une unique fonction objectif. Les problèmes multi-objectifs existent quand un compromis est à rechercher entre plusieurs objectifs contradictoires. Il est éventuellement possible (mais pas nécessairement efficace) de reformuler un problème multi-objectif avec une seule fonction objectif sous forme d'une combinaison des différents objectifs ou en transformant des objectifs sous forme de contraintes.

➤ **Les problèmes d'optimisation avec ou sans contrainte**

Il est important de bien distinguer les problèmes où des contraintes existent sur les variables de décision. Ces contraintes peuvent être simplement des bornes et aller jusqu'à un ensemble d'équations de type égalité et de type inégalité. Il est parfois possible d'éliminer une contrainte égalité par substitution dans la fonction objectif. Naturellement, les problèmes avec contraintes sont plus compliqués à résoudre et utilisent des algorithmes dédiés.

➤ **Les problèmes d'optimisation continue et les problèmes d'optimisation discrète**

Dans certains cas, les variables de décision sont discrètes, le plus souvent sous la forme d'entiers ou de binaires, le problème d'optimisation est dit discret.

Au contraire, dans les problèmes d'optimisation continue, les variables peuvent prendre n'importe quelle valeur, ce sont des réels. Les problèmes d'optimisation continue sont généralement plus simples à résoudre. Un problème d'optimisation mêlant variable continue et variable discrètes est dit mixte.

➤ **Les problèmes d'optimisation déterministe ou stochastique**

Les problèmes d'optimisation déterministe considèrent que les données sont connues parfaitement, alors que dans les problèmes d'optimisation stochastique, ce n'est pas le cas par exemple une approche stochastique peut être pertinente dans le cas où les variables d'un problème sont les ventes futures d'un produit. Dans ce cas, l'incertitude peut être introduite dans le modèle.

2.3 Résolution d'un problème d'optimisation combinatoire

Résoudre un problème d'optimisation combinatoire nécessite l'étude de trois points particuliers :

- La définition de l'ensemble des solutions réalisables,
- L'expression de l'objectif à optimiser,
- Le choix de la méthode d'optimisation à utiliser.

Les deux premiers points relèvent de la modélisation du problème, le troisième de sa résolution. Afin de définir l'ensemble des solutions réalisables, il est nécessaire d'exprimer l'ensemble des contraintes du problème. [2]

2.4 Méthodes de résolution

Souvent les problèmes d'optimisation combinatoire sont des problèmes NP-Complet, ils sont traités de diverses manières, la première manière qui prévaut à toutes les autres est d'essayer de trouver une solution réalisable à ce problème, puis on doit essayer d'améliorer la qualité de ces solutions. Pour ce faire il existe divers algorithmes pour la résolution des problèmes combinatoires, ces algorithmes sont illustrés dans la figure suivante. En décrivant ci-après leurs principales caractéristiques.

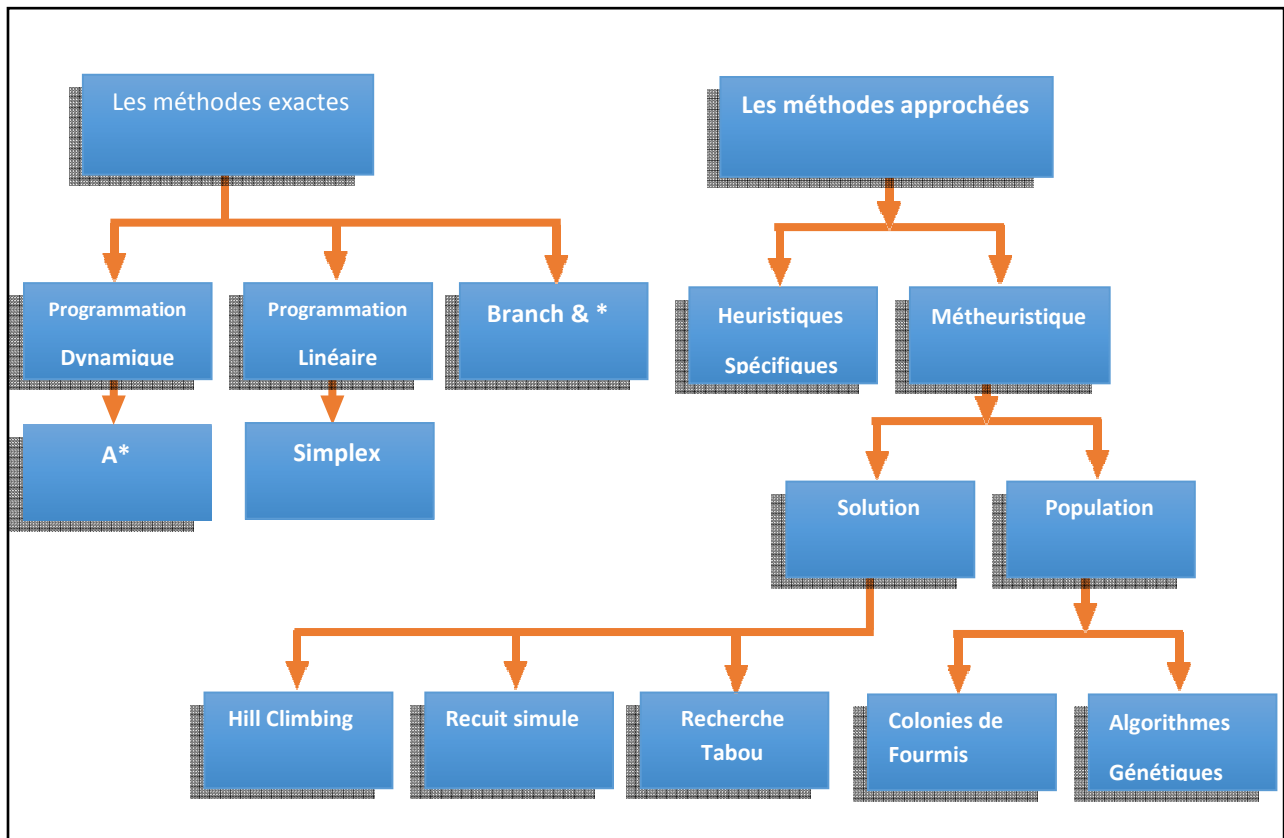


Figure 2: Classification des méthodes d'optimisation combinatoire

2.4.1 Méthodes exactes

Le principe des méthodes exactes consiste à chercher la meilleure solution dans l'ensemble des solutions d'un problème. L'optimisation exacte concerne toutes les méthodes permettant d'obtenir un résultat dont on sait qu'il est optimal à un problème précis. On peut classer les méthodes exactes en trois grandes classes suivantes :

- La programmation dynamique [4],[5].
- La programmation linéaire et quadratique [6],[2].
- Les méthodes de recherche arborescente (branch and bound) [7],[8]

La méthode de branch and bound

La méthode de branch and bound (procédure par évaluation et séparation progressive) consiste à énumérer ces solutions d'une manière intelligente en ce sens que, en utilisant certaines propriétés du problème en question, cette technique arrive à éliminer des solutions partielles qui ne mènent pas à la solution que l'on recherche. De ce fait, on arrive souvent à obtenir la solution recherchée en des temps raisonnables. Bien entendu, dans le pire cas, on retombe toujours sur l'élimination explicite de toutes les solutions du problème.

Pour ce faire, cette méthode se dote d'une fonction qui permet de mettre une borne sur certaines solutions pour soit les exclure soit les maintenir comme des solutions potentielles. Bien entendu, La performance d'une méthode de branch and bound dépend, entre autres, de la qualité de cette fonction (de sa capacité d'exclure des solutions partielles tôt).

➤ Algorithme générale

Début

Placer le nœud début de longueur 0 dans une liste.

Répéter

Si la première branche contient le nœud recherché **alors**

Fin avec succès.

Sinon

- Supprimer la branche de la liste et former des branches nouvelles en étendant la branche supprimée d'une étape.
- Calculer les coûts cumulés des branches et les ajouter dans la liste de telle sorte que la liste soit triée en ordre croissant.

Jusqu'à (liste vide ou nœud recherché trouvé)

Fin

Chapitre I : Optimisation Combinatoire

Programmation Dynamique

La programmation dynamique a été appelée comme cela depuis 1940 par Richard Bellman et permet d'appréhender un problème de façon différente de celle que l'on pourrait imaginer au premier abord. Le concept de base est simple : une solution optimale est la somme de sous-problèmes résolus de façon optimale. Il faut donc diviser un problème donné en sous-problèmes et les résoudre un par un.

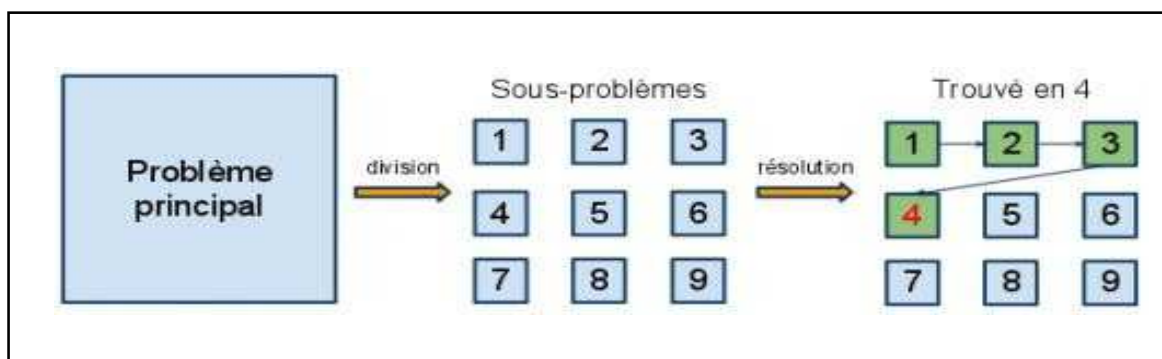


Figure 3: divisé en sous-problèmes

➤ Algorithme général de programmation dynamique

La conception d'un algorithme de programmation dynamique peut être planifiée dans une séquence de quatre étapes.

1. Caractériser la structure d'une solution optimale.
2. Définir récursivement la valeur d'une solution optimale.
3. Calculer la valeur d'une solution optimale en remontant progressivement jusqu'à l'énoncé du problème initial.
4. Construire une solution optimale pour les informations calculées.

2.4.2 Les méthodes approchées ou heuristique

2.4.2.1 Heuristique

Definition

Une heuristique est une méthode généralement utilisée pour résoudre les problèmes de large étendue, non fondée sur un modèle formel et n'aboutit pas nécessairement à une solution optimale. Une méthode heuristique procède par évaluations successives et hypothèses provisoires. Elle constitue une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale.

Types d'heuristiques

Généralement, une heuristique est conçue pour un problème particulier en s'appuyant sur sa structure propre sans offrir aucune garantie quant à la qualité de la solution calculée.

Heuristique constructive

Ce sont des méthodes itératives qui construisent pas à pas une solution. Partant d'une solution partielle initialement vide, elles cherchent à étendre à chaque étape la solution partielle de l'étape précédente, et ce processus se répète jusqu'à ce que l'on obtienne une solution complète, la figure suivante montre le principe de l'approche constructive.

Ces méthodes sont généralement utilisables lorsque la qualité de solution n'est pas un facteur primordial ou la taille de l'instance est raisonnable, en l'occurrence pour générer une solution initiale dans une méta-heuristique, ces méthodes sont rapides et faciles à implémenter.



Figure 4 : Approche constructive

Heuristique de recherche locale

Les méthodes de recherche locale, aussi appelées algorithmes de descente ou d'amélioration itérative, partent d'une solution initiale, et ont pour but de l'améliorer. voir figure la figure ci-dessous.

Le principe général de ces méthodes est le suivant : à partir d'une solution initiale x , dont on connaît la valeur de la fonction objective $f(x)$, on cherche la meilleure solution x_0 dans le voisinage de x . Si l'on ne parvient pas à améliorer x , alors on s'arrête, sinon on remplace x par x_0 , et on recommence.

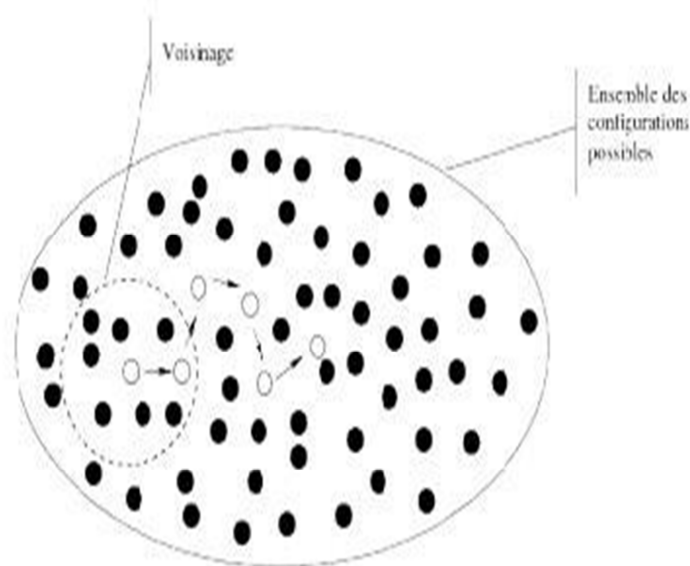


Figure 5 : Approche de recherche locale

2.4.2.2 Méta-heuristiques

Le mot méta heuristique est dérivé de la composition de deux mots grecs :

- heuristique qui vient du verbe heuriskein et qui signifie 'trouver'.
- méta qui est un suffixe signifiant 'au -delà', 'dans un niveau supérieur'.

Les méta-heuristiques sont des méthodes inspirées de la nature, ce sont des heuristiques modernes dédiées à la résolution des problèmes et plus particulièrement aux problèmes d'optimisation, qui visent à atteindre un optimum global généralement enfoui au milieu de nombreux optimums locaux.

Chapitre I : Optimisation Combinatoire

Les méta- heuristiques se subdivisent en deux sous-classes :

Les méthodes de voisinage

Ces méthodes partent d'une solution initiale (obtenue de façon exacte, ou par tirage aléatoire) et s'en éloignent progressivement pour réaliser une trajectoire, un parcours progressif dans l'espace des solutions. Dans cette catégorie, se rangent :

— Le recuit simulé [9].

— La recherche Tabou [10],[11].

- **Le recuit simulé**

La méthode du recuit simulé est une généralisation de la méthode Monte-Carlo ; son but est de trouver une solution optimale pour un problème donné. Elle a été mise au point par trois chercheurs de la société IBM : S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983,

- **La recherche Tabou**

La méthode de recherche tabou, ou simplement méthode tabou, a été formalisée en 1986 par F. Glover [12]. Sa principale particularité tient dans la mise en œuvre de mécanismes inspirés de la mémoire humaine. L'idée consiste à garder la trace du cheminement passé dans une mémoire et de s'y référer pour guider la recherche.

Les méthodes évolutives

Ces méthodes se distinguent de celles déjà étudiées par le fait qu'elles opèrent sur une population de solution ; pour cela, elles sont souvent appelées des méthodes à base de population. Certaines d'entre elle ont des principes inspirés de la génétique et du comportement des insectes ; on trouve entre autre : les algorithmes génétiques, les colonies de fourmis [13], les essaims particulaires [14],[15]... etc. La complexité de ces phénomènes biologiques ont servi de modèle pour des algorithmes toujours plus sophistiqués ces vingt dernières années.

3. Éléments de la théorie de la complexité

La théorie de la complexité traite de la difficulté de résolution des problèmes formulés mathématiquement. Cette difficulté est mesurée ainsi bien en temps (temps nécessaire à sa résolution) qu'en espace (espace mémoire). Il est prévisible d'avoir une augmentation de ces quantités en fonction de la taille du problème, on s'intéresse alors à la manière dont elle augmente. La pire situation étant que le temps nécessaire à la résolution d'un problème soit une fonction exponentielle de la taille du problème. [16]

3.1 Notions de base

L'évaluation de la complexité en temps des algorithmes repose sur une définition précise du temps qui ne dépend pas de la vitesse d'exécution de la machine ni de la qualité du code produit par le compilateur. En effet, le temps de calcul obtenu par l'exécution d'un programme sur une machine donnée ne reflète pas réellement la difficulté de celui-ci, donc ne devient pas plus facile lorsqu'un processus plus performant est sur le marché.

Généralement, pour aborder l'étude des problèmes d'optimisation combinatoire il est nécessaire de se familiariser avec les termes suivants :

Problème de décision [16]

C'est un problème qui possède pour solution Oui/non. On n'en déduit qu'un problème décisionnel abstrait fait correspondre à toute instance du problème l'ensemble des solutions vrai, faux.

Instance d'un problème [16]

soit X un problème caractérisé par l'ensemble E de ses données,
 $E = (e_1, e_2, \dots, e_n)$. Une instance de X serait alors caractérisée par un ensemble concret $E' = (e'_1, e'_2, \dots, e'_n)$. Une deuxième instance serait caractérisée par un deuxième ensemble concret $E'' = (e''_1, e''_2, \dots, e''_n)$ et ainsi de suite.

L'invariant Suivant est toujours vérifié: $|E| = |E'| = |E''|$.

Chapitre I : Optimisation Combinatoire

Problème d'optimisation [16]

Un problème d'optimisation est un problème caractérisé par un quadruplet (I, S, f, m) , où I représente les données du problème, S est le type de collection /ensemble ou le type d'éléments qu'on attend comme résultat, f est la fonction qui mesure la qualité du résultat et m (min ou max) indique s'il s'agit de minimisation ou maximisation de la taille de ce résultat.

Réductibilité [16]

Un problème X peut être ramené (réduit) à un autre problème X_0 si une instance quelconque de X peut être facilement reformulée comme instance de X_0 dont la solution sera aussi solution pour X .

Réductibilité en temps polynomial [16]

La fonction qui réduit le problème X en X_0 peut être obtenue en temps polynomial.

3.2. Classes de Complexité [16]

La théorie de la complexité des algorithmes étudie formellement la différence intrinsèque des problèmes algorithmiques et il existe un très grand nombre de classes de complexité, elles ont été introduites pour classer les algorithmes selon leurs caractéristiques, on définit les trois grandes classes P, NP et NP-Complexe.

➤ **La classe P**

Il s'agit de la classe de complexité contenant les problèmes décisionnels pouvant être résolus par une machine de Turing déterministe pendant un temps de calcul polynomial. La complexité en temps de calcul pour un problème X exprime un rapport entre la taille de ses données et le temps utilisé pour le résoudre, pour cette raison qu'on dit que ces problèmes appartenant à cette classe comme étant efficaces ou traitables, bien qu'il soit naturel et raisonnable de considérer comme intraitables, dans le cas pratique, pour un problème qui demande un temps de l'ordre de grandeur aussi élevé. Cette classe est réputée contenir de nombreux problèmes naturels, comme le calcul du plus grand diviseur commun.

➤ La classe NP

La classe des problèmes NP regroupe l'ensemble de problèmes qui peuvent être résolus avec une machine de Turing non déterministe et admettent un algorithme polynomial pour tester la validité d'une solution du problème traité (i.e. il est possible de vérifier que la solution proposée est correcte en un temps polynomial par rapport à la taille du problème).

➤ La Classe NP-Complet

Un problème de décision π est NP-Complet s'il satisfait les deux conditions suivantes : π appartient à NP, et tout problème NP se réduit à π en temps polynomial [36]. La classe NP-complet contient un ensemble de problèmes les plus difficiles de NP, de tels problèmes apparaissent régulièrement. Citons entre autres, le problème du voyageur de commerce, celui du stable maximal, le calcul du nombre chromatique...etc. tous ces problèmes correspondent à une difficulté pratique pour trouver un algorithme efficace pour de très grands graphes (les seuls algorithmes déterministes connus étant de complexité non polynomiale, souvent exponentielle).

Cette difficulté n'empêche pas de mettre au point des heuristiques, c'est-à-dire qui n'assurent pas à coup sûr l'obtention d'une solution du problème, mais y parviennent assez souvent ou assez près. Après avoir utilisé une méthode heuristique, on peut conclure la recherche avec une méthode exhaustive, c'est-à-dire d'examen a priori des toutes les configurations plausibles.

La grande question ouverte dans ce domaine, et qui est classée par l'institut de mathématique Clay parmi les 7 problèmes du prix du millénaire, concerne l'inclusion des classes P et NP, la question est traditionnellement formulée ainsi : <Est-ce que $P = NP$ ou $P \neq NP$? >.

Cette question n'a pas encore trouvé de réponse car de très nombreux problèmes fondamentaux s'avèrent être < NP -difficiles >, et aucun algorithme polynomial n'existe pour les résoudre, sauf si $P=NP$. Puisqu'il est tout de même important de pouvoir les résoudre, les scientifiques introduisent et utilisent diverses méthodes pour attaquer ces problèmes.

4 . Exemples de problème d'optimisation combinatoire

4.1 Problème du sac à dos

Le problème du sac à dos (en anglais Knapsack problem), modélise une situation analogue au remplissage d'un sac a dos, ne pouvant supporté plus d'un certain poids, avec tout ou une partie d'un ensemble donné d'objet ayant chacun un poids et une valeur. Les objets mis dans le sac a dos doivent maximiser la valeur totale, sans dépasser le poids maximum.

4.2 Problème de transport

Le problème du transport est un programme de minimisation de cout de transport. Il a pour but d'acheminer au moindre des marchandises depuis m origines vers n destinations. Autrement dit, le problème peut s'illustrer comme suit : partons des places A_i (dépôts, usines) ayant des offres des produit x_{ij} , et d'autres places B_j (points de vente, client), qui expriment des demandes sur les produits x_{ij} ,(x_{ij} représente la quantité des produits transportés de l'origine A_i vers la destination B_j) La formulation et la résolution d'un problème de transport a pour cible :

- La minimisation du cout de transport.
- L'obtention d'une demande excédentaire nulle c'est-à-dire satisfaire la demande dans toutes les places B_j et éliminer les fournitures dans toutes les place A_i .

5. Conclusion

Nous avons présenté dans ce chapitre quelques méthodes les plus connues pour la résolution des problèmes combinatoires. Ces méthodes sont généralement classées en deux catégories : les méthodes exactes et les méthodes approchées. Les méthodes exactes ont l'avantage de garantir l'obtention de la solution optimale. Cependant, elles présentent un inconvénient majeur qui est celui du temps d'exécution important, car ces méthodes parcourent tout l'espace de recherche. Les méthodes approchées sont des algorithmes qui tendent à s'approcher de l'optimal en temps raisonnable. Ces algorithmes présentent l'avantage d'être rapides, mais ne garantissent pas l'obtention de la solution optimale. Néanmoins, ils sont perçus par un bon nombre de chercheurs comme étant un bon compromis entre le temps d'exécution et la qualité de la solution

Chapitre II

Le Problème de Bin Packing

1. Introduction

Le problème de bin packing consiste d'une manière générale à trouver le rangement le plus économique possible pour un ensemble d'objets dans des boîtes dites « bins ».

Ainsi, les problèmes de type bin packing consistent à placer des articles caractérisés par leur formes dans une ou plusieurs boîtes. Les variantes se distinguent selon la dimension, la connaissance a priori des articles, la forme des articles et des boîtes (carré, rectangulaire, circulaire...), la possibilité de modifier l'orientation des articles. Elles se distinguent également par le problème de faisabilité, c'est-à-dire, savoir s'il existe un rangement réalisable des articles dans les boîtes, par exemple la minimisation du nombre de boîtes (bin packing) ou la minimisation des dimensions d'une seule boîte (hauteur - strip packing, aire - rectangle packing, volume...), ou encore la maximisation de la valeur du rangement (problèmes de sac à dos – knapsack problems).

2. Domaines d'application du problème de bin packing

Le secteur industriel est fréquemment confronté à des problèmes de découpe de matières premières. Dans l'intérêt de réduire les coûts et impacts environnementaux, l'objectif est de minimiser la quantité de matériaux perdue dans les chutes. Dans le secteur de transport, le développement des activités logistiques provoque la multiplication de problèmes de conditionnement. La résolution de ces problèmes est un enjeu économique très important pour de nombreuses entreprises, dont ci-dessous quelques exemples d'application:

- Dans la fabrication des tubes de tailles différentes. Au lieu d'acheter plusieurs machines pour produire des tubes de tailles différentes, une seule machine produit des tubes de tailles fixées et une seconde machine découpe les tubes en petits tubes de tailles différentes. Dans ce cas, le problème bin packing est utilisé pour découper plus efficacement afin de minimiser le coût de fabrication. Différents logiciels industriels pour la découpe des tubes, découpe 2D/3D pour le découpe automatique sont largement commercialisés.

Chapitre II: Le Problème de Bin Packing

- L'emballage de meubles dans des camions de charge limite peut être réglé par l'utilisation du problème bin packing qui a pour objectif d'utiliser le moins de cabines ou camions possible pour emballer tous les éléments sans excéder la charge des cabines et camions.

- La réalisation d'un ensemble de tâches à faire: chaque tâche doit être exécutée dans un temps quelconque. Des machines sont utilisées pour réaliser ces tâches dans un délai fixé dans le but de minimiser les machines utilisées sans dépasser le délai.

3. Contraintes pratiques et classification

De nombreux problèmes pratiques se modélisent sous la forme d'un problème de bin packing. Cependant, chaque problème réel présente ses propres spécificités telles que

➤ les caractéristiques propres aux objets

- objets de formes homogènes ou non homogènes.
- objets de tailles uniformes ou différentes.
- objets déformables ou non déformables,...etc.

➤ les spécificités propres au problème

- nombre de dimensions du problème.
- disposer d'un seul bin (problème de décision ou problème de maximisation).
- chercher à minimiser le nombre de bins à utiliser.
- chercher à minimiser la surface ou le volume global des objets à placer,...etc.

➤ les contraintes propres au problème

- contraintes d'équilibre entre les objets,
- contraintes d'orientation d'un objet,
- contraintes de poids par exemple, le poids d'un bin complet ne peut pas excéder une limite donnée),
- contraintes de placement, certains objets très lourds doivent être placés en bas, d'autres fragiles doivent être placés en dessus,...etc.

Dyckhoff et Finke [17] ont proposé une typologie qui permet d'organiser les problèmes de découpes et de placements en tenant compte de quatre caractéristiques principales :

Chapitre II: Le Problème de Bin Packing

- le nombre de dimensions du problème;
- le type de tâche : tous les objets et une sélection de bins, ou bien une sélection d'objets et tous les bins;
- les caractéristiques des bins : 1 seul bin, des bins de tailles identiques, ou bien des bins de tailles différentes;
- les caractéristiques des objets : objets identiques, peu d'objets de formes différentes, plusieurs objets de formes différentes ou bien des objets de formes relativement identiques.

Une typologie plus récente a été proposée par Wäscher et al. [18] dans le but d'inclure les problématiques de découpe et de rangement, et d'établir une catégorisation complète de tous les problèmes connus dans le domaine.

En ce qui concerne le problème de bin packing en deux dimensions (BPP-2D), les deux spécificités les plus rencontrées sont les suivantes :

- ❖ l'orientation : les objets peuvent être à orientation fixe (le cas orienté) ou bien ils peuvent être tournés de 90 degrés (le cas non orienté).
- ❖ la contrainte guillotine : Si elle est imposée, les objets rangés peuvent être restitués par des coupes bout à bout parallèles aux dimensions de bins.

4. Le problème de bin packing uni- , bi- et tri-dimensionnel

Les problèmes de bin packing se distinguent, selon la dimension, en :

Bin packing uni-dimensionnel, où les objets sont caractérisés par une seule mesure comme la hauteur, la largeur ou le poids, qu'il faut ranger sur un axe de taille fixée. Exemple : le rangement de fichiers sur un support informatique, la découpe de câbles, le remplissage de camions ou de containers avec comme seule contrainte le poids ou le volume des articles.

Bin packing bi-dimensionnel, où il faut ranger les objets sur une surface limitée, tels que la découpe de matière première (bois, verre, acier, etc.), le placement de boîtes sur une palette ou le placement des articles dans un journal.

Bin packing tri-dimensionnel, où les objets sont rangés dans un espace de volume fixe comme le rangement d'objets dans un entrepôt, des camions, etc.

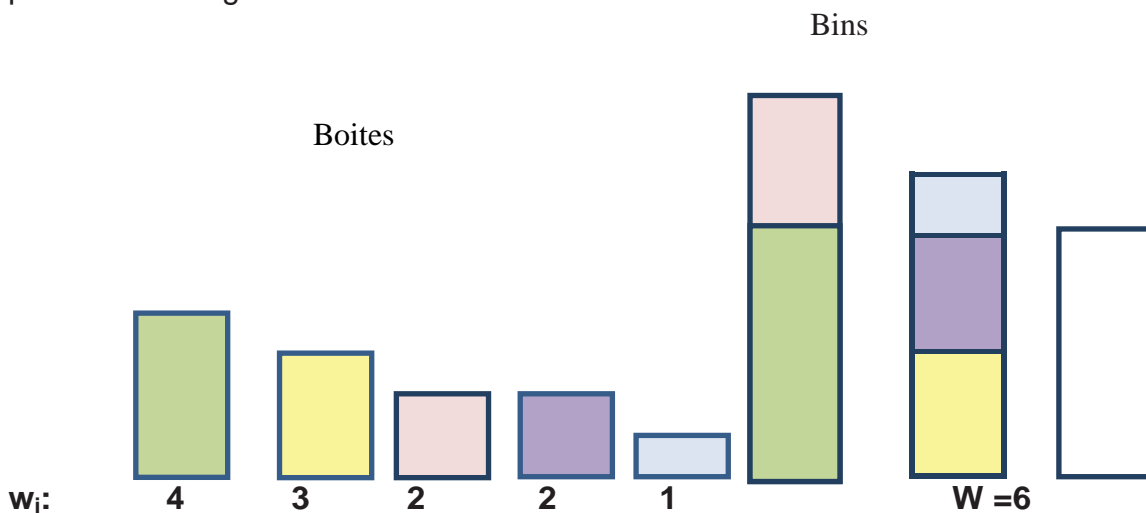
4.1 Le problème de bin packing uni-dimensionnel

Le problème de bin packing uni-dimensionnel (BPP1D) consiste à minimiser le nombre de bins uni-dimensionnel nécessaires pour ranger une Liste d'objets caractérisés par leur longueur.

Ce problème est NP-complet[3], Une instance de BPP-1D, notée (I, w, W) , comprend un ensemble $I = \{1, 2, \dots, n\}$ de n objets et une fonction w qui associe à chaque objet i une valeur w_i qui correspond à sa longueur, et W une valeur positive représentant la taille du bin.

Exemple

Placer un ensemble de boîtes de longueurs différentes dans le moins de bins possible de longueur fixée.



Données

N : ensemble de n boîtes avec w_i longueur de la boîte ($i \in N$) et W longueur des bins

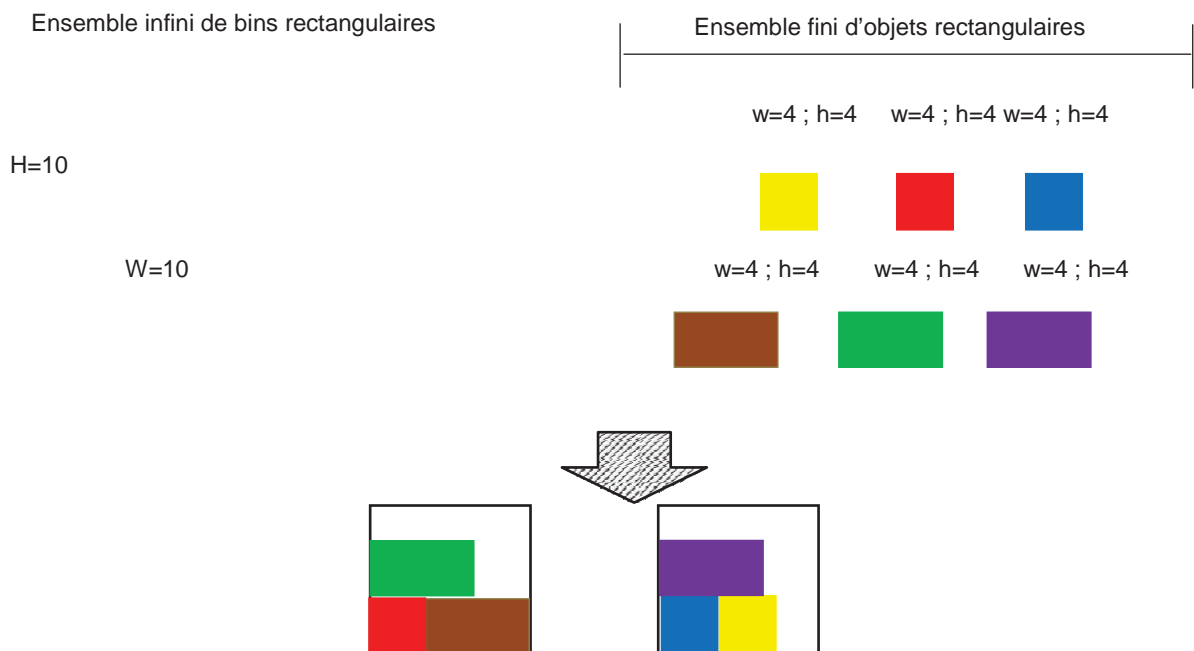
Objectif : minimiser le nombre de bins utilisés

Contraintes: ne pas dépasser la capacité d'un bin.

4.2 Le problème de bin packing bi-dimensionnel

Le problème de bin packing bi-dimensionnel (BPP-2D) est une généralisation naturelle de BPP-1D. Il s'agit de minimiser le nombre de grands rectangles (bins) identiques pour ranger une liste d'objets rectangulaires. Les objets doivent être rangés de telle manière que les côtés des rectangles soient parallèles à ceux du bins. On note (l, w, h, W, H) une instance de BPP-2D,

$l = \{1, 2, \dots, n\}$ est la liste des objets à ranger, et une fonction w (Respectivement h) qui associe à chaque objet i une valeur w_i (respectivement h_i) qui correspond à sa longueur (respectivement sa largeur), W et H représentent la longueur et la largeur du bin.[3]



Chapitre II: Le Problème de Bin Packing

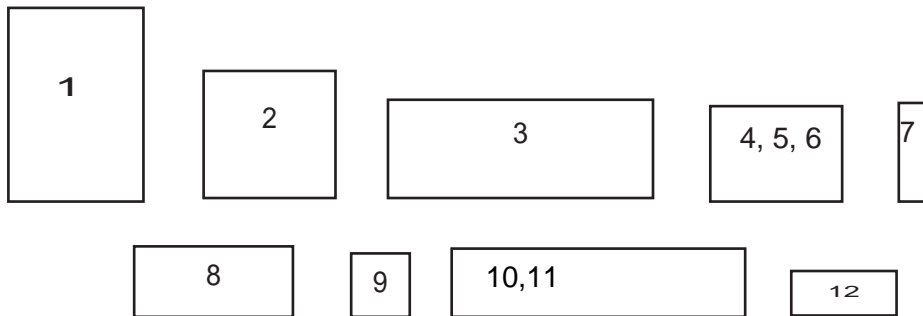
La différence entre BPP-1D et BPP-2D réside dans le problème de faisabilité car étant donné un ensemble d'objet et un bin, il s'agit de déterminer s'il existe un placement réalisable pour ces objets dans le bin. Le problème est trivial en une dimension (il suffit de sommer la longueur des objets) et NP-complet en deux dimensions.

Deux types de problèmes imbriqués sont considérés : un problème d'affectation et plusieurs problèmes de faisabilité. La question d'orientation des objets se pose pour le BPP- 2D.

Exemple

Soit l'instance BPP-2D suivante :

$I = (A, B) = \{ A = \{ a_1 = (4, 8), a_2 = (5, 4), a_3 = (3, 10), a_4 = (3, 5), a_5 = (3, 5), a_6 = (5, 3), a_7 = (3, 1), a_8 = (7, 2), a_9 = (2, 2), a_{10} = (11, 2), a_{11} = (2, 11), a_{12} = (3, 1) \}, B = (12, 10). \}$



Solution

On place les objets en respectant l'orientation

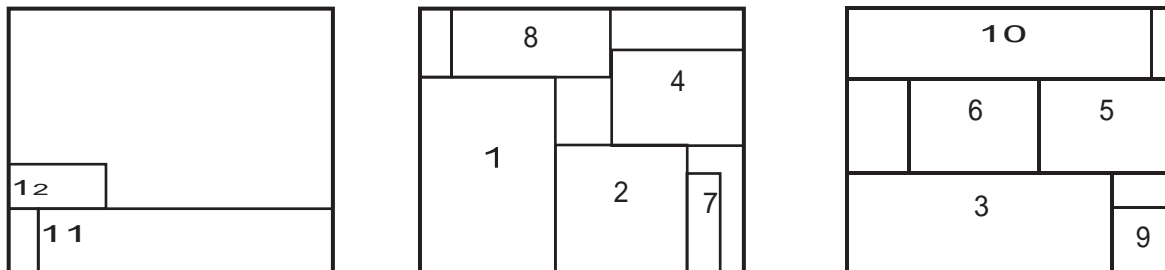


Figure 6. Bin packing le cas orienté.

Solution sans orientation

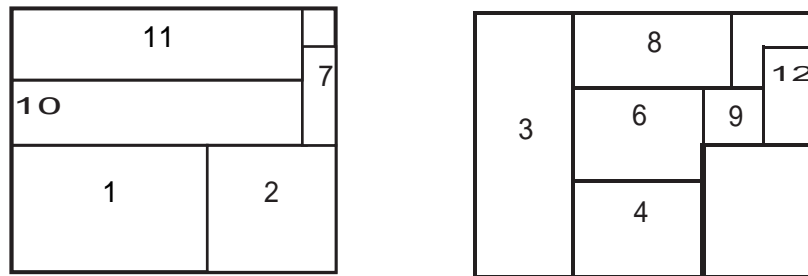


Figure 7. Bin packing le cas non orienté.

Un exemple de solution de cette instance dans le cas orienté est donné dans la figure 6. La figure 7 représente une solution pour la même instance, mais en considérant cette fois-ci le cas non orienté. Cette solution est optimale.

4.3 Le problème de bin packing tri-dimensionnel

Le problème de bin packing tri-dimensionnel (BPP-3D) connu aussi sous le nom de problème de chargement de conteneurs, est un problème combinatoire dont dérive de nombreuses applications industrielles [18]. Ce problème consiste à placer un ensemble S de n objets de dimensions (w_i, h_i, d_i) , $i=1, \dots, n$, dans un nombre minimum de conteneurs identiques R_j , $j=1, \dots, m$, de dimensions (W, H, D) où W (resp. w_i) est la largeur, H (resp. h_i) la hauteur et D (resp. d_i) la profondeur des conteneurs (resp. des objets) tout en respectant certaines contraintes. [19]

Exemple

Une instance est caractérisée par un ensemble de n objets (petits parallélépipèdes) et un ensemble de supports (grands parallélépipèdes ou des palettes).

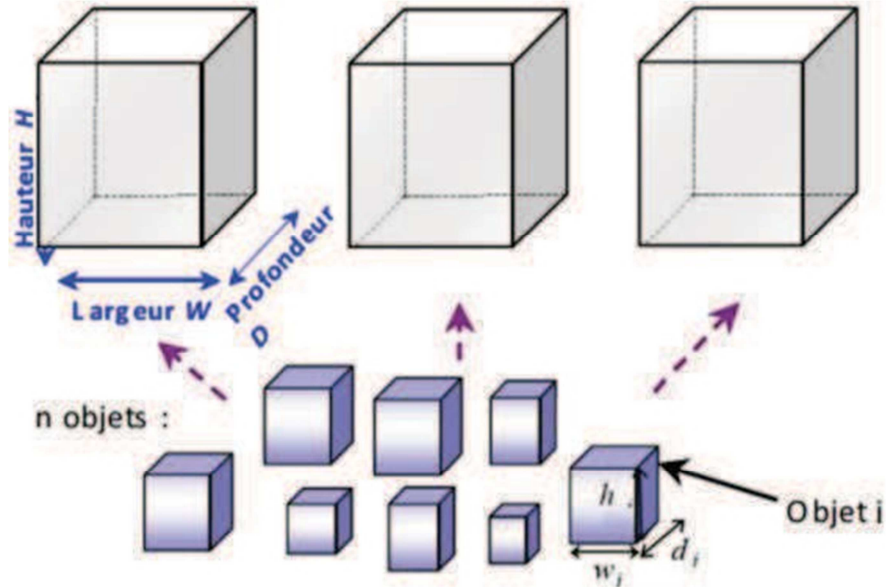


Figure 8. Illustration graphique du BPP-3D.

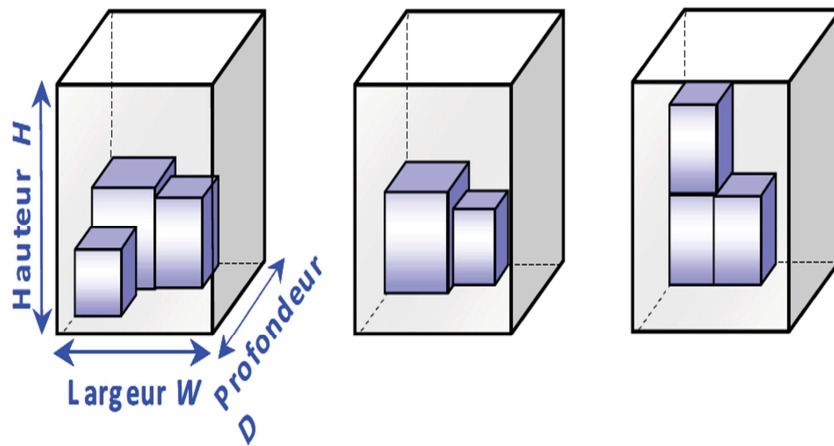


Figure 9. Illustration d'une solution réalisable pour BPP-3D.

Il s'agit d'un problème lié au problème de chargement, des contraintes supplémentaires sont imposées sur chaque chargement :

- (i) éviter le débordement,
- (ii) respecter le tonnage (volume intérieur) du support,
- (iii) et définir des précédences et des préférences de regroupement entre certains objets.

Le problème de base revient donc à placer tous les objets de sorte que:

- les objets ne se chevauchent pas.
- le nombre de supports utilisés soit minimum.
- les contraintes soient respectées.

Chapitre II: Le Problème de Bin Packing

Dans notre mémoire notre intérêt sera porté sur les problèmes de placement, nous développerons alors les notions de bases de cette théorie.

Chapitre III

Le placement dans l'industrie

1. Introduction

La réduction des coûts de production est l'un des problèmes majeurs dans les industries manufacturières dans un monde de plus en plus concurrentiel où les produits sont disponibles presque instantanément partout dans le monde. La forte utilisation des matériaux présente un intérêt particulier pour les industries de production de masse, car de petites améliorations de la disposition peuvent entraîner des économies importantes de matériau et réduire considérablement les coûts de production. Des problèmes de découpe et de placement sont rencontrés dans de nombreuses industries. La complexité du problème et l'approche de la solution dépendent de la géométrie des éléments à placer et des contraintes imposées.

Alors que l'industrie du bois, du verre et du papier est principalement concernée par la coupe de figures régulières, des objets de forme irrégulière et arbitraire doivent être placés dans l'industrie de la construction navale, du textile et du cuir.

2. Notions générales sur les problèmes de placement

Le placement fait partie des problèmes d'optimisation combinatoire qui suscitent beaucoup d'intérêt. Rencontré chaque fois que la fabrication d'un objet manufacturé est réalisée par transformation de la matière.

La fabrication d'un objet est souvent conduite avec l'objectif d'une réduction du coût de revient à travers la réduction de la consommation en matière. Cette tendance est en particulier liée aux augmentations de prix des matières premières. Aussi, en général, l'exploitation optimale des ressources constitue une préoccupation croissante dans les industries manufacturières. Une formulation industrielle du problème de placement en deux dimensions peut se présenter comme suit : « Étant donné une matière première présentée sous forme de plusieurs unités de dimensions et formes éventuellement différentes, comment produire une quantité de pièces, en fonction de la demande et des niveaux de stocks, en utilisant le minimum de matière, et ce dans un temps compatible avec les délais fixés par le client ». La quantité de pièces à produire peut être connue ou non à l'avance. Si l'objectif principal consiste à minimiser la consommation en matière première, il est nécessaire de produire vite pour satisfaire des délais de livraison impératifs et de plus en plus courts, au « juste-à-temps » pour éviter les frais de stockage.

Le problème de placement a été étudié pour la première fois par Kantorovick [13] dont le but était de le modéliser sous une forme cohérente. Gilmore et Gomory [21] ont repris ces travaux pour la résolution de quelques unes de ces variantes et les ont généralisés aux problèmes de découpe à deux dimensions.

Le problème de découpe dépend des éléments suivants :

Le support initial

Le support est l'un des éléments de base du problème de découpe. les supports diffèrent entre eux par leur :

Forme géométrique

- Si la forme du support est circulaire, il est appelé cercle initial de dimension (R_a) où R_a est son rayon.
- Si le support est de forme rectangulaire, il est appelé dans ce cas rectangle initial, de dimensions (J, W) où J représente la longueur et W sa hauteur.

Dans les deux cas on parle de problème de découpe à deux dimensions.

La longueur (ou la hauteur) est négligée si elle est très grande par rapport aux dimensions des pièces à produire. Dans ce cas le problème peut être représenté par une bande, on parlera donc de problème de découpe à une dimension.

Un problème de découpe est à trois dimensions si son support est caractérisé par une base rectangulaire de dimension (J, W) et d'une épaisseur E .

Homogénéité et régularité

Un support est homogène si sa surface est partout la même et il est régulier s'il ne présente aucun défaut.

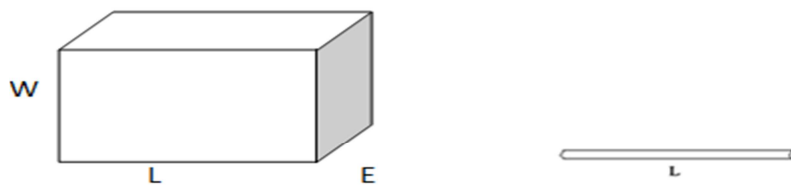
Dissymétrie :

Sur certain support comme les tissus à motifs, les pièces à découper ont souvent une orientation imposée, on parlera alors de pièce fixée.



i. Problème de découpe à deux dimensions : (a) rectangle.

(b) cercle



ii. Problème de découpe à trois dimensions.iii. Problème de découpe à une dimension.

Les différents types de supports

Les pièces à découper

Les caractéristiques principales d'une pièce à découper sont :

- Sa forme géométrique.
- le respect des contraintes liées au support et au coût de production.

Les pièces sont dites fixées si l'orientation est imposée si non les rotations sont permises et les pièces peuvent être pivotées de 90° .

3. Types de problèmes de placement

Les problèmes de placement se produisent dans divers domaines d'application impliquant des contraintes et des objectifs différents. Dans la suite, certains des problèmes les plus importants sont brièvement définis. Des problèmes industriels plus importants peuvent également apparaître en tant que combinaisons de deux ou plusieurs de base. [5].

- **Problème de stock de découpe**

Le problème de stock de découpe concerne la découpe de pièces (composants) d'une liste de commande donnée à partir d'un ensemble de feuilles de stock (contenants). Ce problème peut être scindé en deux sous-problèmes, un problème d'assortiment (détermination des feuilles à garder en stock) et un problème de perte de finition (détermination d'algorithme de coupe pour minimiser le gaspillage).

- **Problème de perte de finition**

Le problème de perte de finition concerne l'allocation de la liste de commande sur les feuilles de stock données. La liste de commandes décrit l'ensemble des pièces qui doivent être placées. L'objectif est de minimiser le coût total des feuilles de stock nécessaires pour exécuter la commande.

- **Problème d'assortiment**

Ce problème implique la détermination des tailles de stock nécessaires pour remplir la liste de commandes. La liste de commande doit être affectée à une fourniture de feuilles de stock de sorte que la meilleure sélection de feuilles soit utilisée.

- **Problème de chargement**

Le problème de chargement décrit le processus d'ajustement d'un nombre maximum de boîtes sur une palette ou dans un conteneur. Le problème de chargement des palettes peut être considéré du point de vue du fabricant, où des boîtes identiques doivent être chargées sur une palette (problème de chargement des palettes du fabricant), ainsi que du côté du distributeur où la palette doit être emballée avec des articles non identiques (problème de chargement des palettes du distributeur). Le chargement des conteneurs est similaire au chargement des

Chapitre III :Le placement dans l'industrie

palettes, bien que dans les applications pratiques, les deux variantes du problème de chargement puissent être distinguées par leurs contraintes.

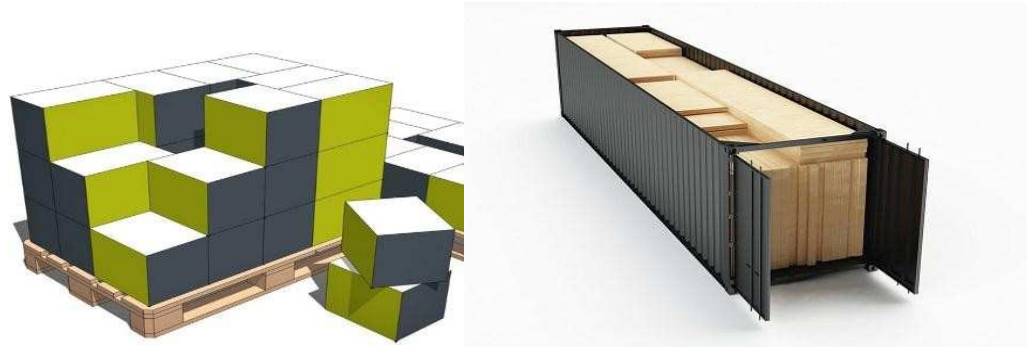


Figure10: Chargement d'une palette et un conteneur

4. Propriétés matérielles

Une variété de matières premières sont disponibles dans l'industrie, on peut citer la tôle le textile et le cuir, ces matériaux sont disponible sous la forme de feuille avec des dimensions fixes ou en tant que matériau enroulé avec une largeur fixe. Cela a un impact sur l'objectif de la tâche de placement. Dans le premier cas, la forme d'un reste possible décide s'il peut être utilisé dans une future tâche de placement ou doit être traité comme du gaspillage. Pour les matériaux enroulés, cela n'est généralement pas une contrainte.

- **Le placement dans l'industrie de la tôle**

L'industrie des tôles doit faire face à des problèmes de placement réguliers et irréguliers. Certaines contraintes concernant les propriétés du matériau, le processus de coupe et les aspects de planification distinguent la tâche de placement dans ce domaine par d'autres industries. En plus de réduire le gaspillage au minimum, il existe un certain nombre d'autres facteurs qui décident de la disposition finale des pièces [11].



Figure 11:Le placement dans l'industrie de Tôle

- **Le placement dans l'industrie textile**

L'industrie textile utilise habituellement un matériau enroulé est donc principalement concernée par le problème de placement des bandes.

Les propriétés du matériau limitent la génération de disposition. Comme le tissu a souvent certaines propriétés directionnelles et un motif, l'orientation des pièces est généralement limitée à des intervalles de rotation de 0 °et 180 °. Dans de nombreux cas, il peut ne pas être possible d'orienter les pièces car le tissu a des propriétés différentes.[11].



Figure 12:Le placement dans l'industrie de textile

- **Le placement dans l'industrie du cuir**

La tâche de placement dans l'industrie du cuir est très complexe car les pièces à placer ainsi que les objets, les peaux, sont très irréguliers. Comme le cuir est un matériau naturel, par opposition aux matériaux manufacturés utilisés dans l'industrie du textile et de la tôle, les peaux sont constituées de surface ayant diverses qualités. La différence de qualité peut être

due à des défauts et des différences de couleur.

Le processus de placement doit donc faire correspondre les parties avec leurs surfaces de qualité respectives sur la peau. Habituellement, le traitement d'image a lieu avant le processus placement pour déterminer la forme et la qualité des peaux. Comme cette tâche de placement est très complexe et doit tenir compte de nombreuses exigences spécifiques, des emballages de placement spécialement conçus sont disponibles pour l'industrie du cuir.

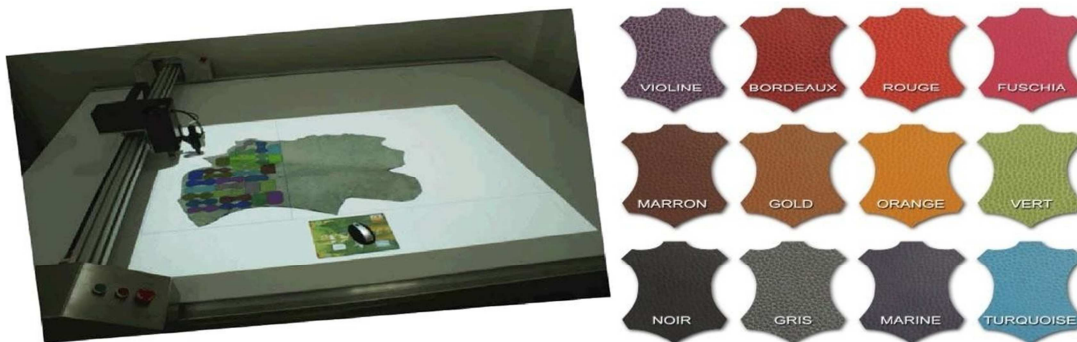


Figure 13:Le placement dans l'industrie de cuir

5. Processus de placement

Les parties à placer peuvent contenir des surfaces vides, dont certaines peuvent être suffisamment grandes pour être prises en compte pour placement d'éléments plus petits. Cette technique est appelée emboîtement dans le trou et est très commune dans l'industrie de la construction navale.

Pour réduire le gaspillage, l'algorithme de placement doit être capable de suivre et de placer dans des surfaces vides de formes rectangulaires. Parfois, la tâche de placement actuelle ne contient pas un nombre suffisant de formes relativement petites. Comme la matière première est souvent trop précieuse pour être gaspillée, certaines pièces de remplissage peuvent être désignées et utilisées à la place. Ceux-ci ne font pas partie de la commande en cours et peuvent donc ne pas être requis immédiatement, mais sont produits pour le stock.

Les tâches de placement plus importantes peuvent impliquer des matériaux de types différents, par ex. épaisseur. Dans l'industrie de la construction navale, des feuilles d'épaisseurs différentes peuvent être impliquées dans le processus de placement . Alors qu'un certain nombre de pièces nécessitent un certain type de feuille, souvent plusieurs épaisseurs conviennent pour une sous-section de la liste de commande.

Par conséquent, en fonction de la disponibilité, l'algorithme de placement doit décider de la meilleure allocation .

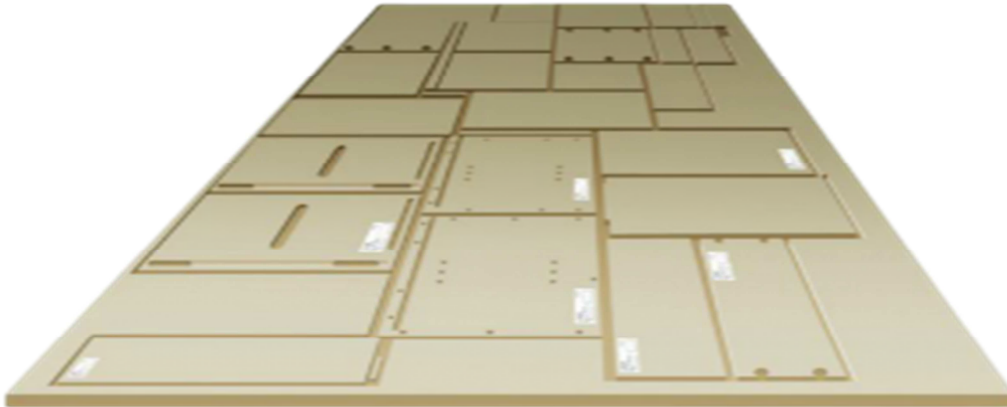


Figure 14:Le placement des pièces irrégulières selon un algorithme de placement

6. Planification (séquencement)

La séquence dans laquelle les pièces sont coupées peut être importante pour le processus de fabrication ultérieur. C'est le cas lorsque les pièces doivent être traitées dans différentes étapes. Si les dispositions sont grandes, une allocation spéciale des composants (pièces) par rapport aux feuilles facilite cela. La séquence des pièces peut également être importante pour le placement et l'expédition. Des contraintes géométriques ou de poids peuvent exiger que les pièces soient placées dans un certain ordre. Parfois, des listes de commandes différentes sont placées dans un contenant (feuille) afin de maximiser l'utilisation du matériel. D'où la séquence d'ordre des pièces joue également un rôle dans l'expédition.

7. Processus de coupe

La technique de coupe utilisée pour obtenir les pièces a un grand impact sur la génération de la mise en page. En fonction de la technologie de découpe (découpe laser et plasma, estampage, par exemple), une distance minimale entre les pièces est requise.

Ce paramètre est appelé largeur de pont. En découpe laser et plasma, le procédé fonctionne avec une certaine largeur. Afin de ne pas endommager les pièces, une certaine distance entre les formes voisines est nécessaire. Dans les processus d'estampage, le matériau a tendance à glisser sur les arêtes de coupe si la largeur du pont est trop petite. Un autre paramètre important qui détermine le processus de coupe est la longueur de coupe. Les contenants (feuilles) peuvent être optimisés pour que la découpe de toutes les pièces puisse être réalisée en minimisant la distance totale. [21].



Figure 15:Technique de coupe

8. Les différents types de découpe

En pratique, les utilisateurs sont souvent contraints par le matériel de découpe disponible, la façon de procéder pour la réalisation d'un plan de découpe est différente. [23]

- **La découpe guillotine**

Si on suppose que le support est une plaque rectangulaire, alors la découpe est effectuée par la dissection en allant d'un côté à son opposé parallèlement aux deux autres.



Figure 16 : Exemple du coupe guillotinable

- **La découpe non guillotine**

Le terme «non guillotine» se réfère au fait que les coupes ne se limitent pas aux coupures de guillotine imposées par certaines machines à couper (CNC) ou des environnements de placement (une guillotine orthogonale coupée sur un rectangle) est une coupe d'un bord du rectangle au bord opposé, parallèle Au bord restant

En générale cette découpe engendre une solution meilleure que celle réalisée par les découpes du type guillotine, en effet cette découpe consiste à utiliser le même procède que dans la découpe Guillotine, et de plus, elle peut être effectuée tout en marquant des arrêts d'atteindre le côté opposé du (sous) rectangle à découper.

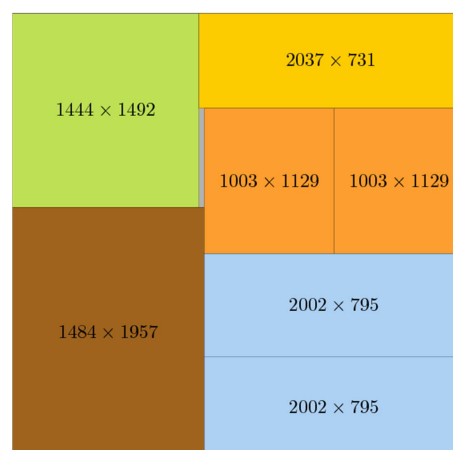


Figure 17 : Exemple non-guillotinable

- **La découpe non orthogonale**

Cette découpe ne prend pas en considération l'orientation des pièces. Cette fois les pièces peuvent être pivotées et translattées (on exécute des rotations sur les pièces, donc elles ne sont pas fixées).

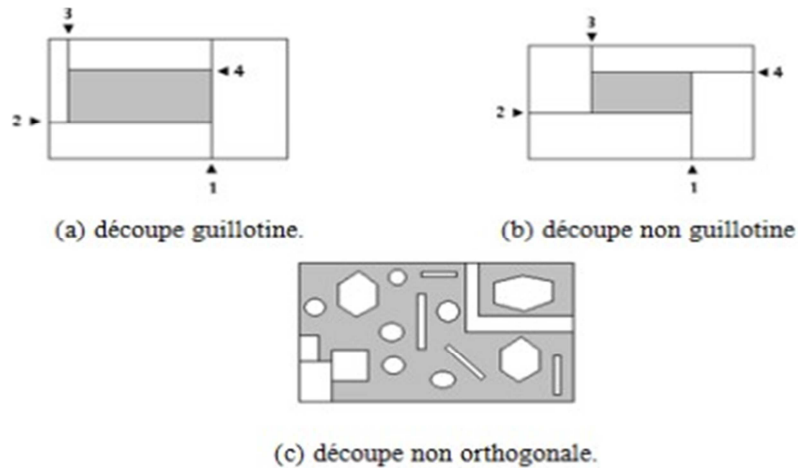


Figure 18:Les différents types de découpe

9. Industries avec des problèmes de placement rectangulaires

Le groupe des problèmes de placement rectangulaires apparaît dans l'industrie du papier, du bois et du verre. La technique de coupe, qui implique des coupes de cisaillement, impose une contrainte caractéristique sur les aménagements dans cette surface.

Les motifs de placement doivent être guillotine, de sorte que les pièces puissent être obtenues par coupe droite à travers la disposition restante seulement.

Le placement dans l'industrie du verre et du bois doit également tenir compte de diverses gammes de qualité et des défauts de la matière première. Des tâches de placement rectangulaires peuvent se produire en tant que problèmes des placements de bande et de bac. [5].



Figure 19 : les pièces placées dans l'industrie de *bois*

- **Le placement de Bande (strip packing)**

L'industrie du papier est principalement concernée par le problème de placement des bandes, car la matière première est disponible sous forme de rouleaux. D'où le processus de placement vise à réduire la hauteur de contenant.



Figure 20:Le placement de bande dans l'industrie de papier

- **Le placement de bac (Bin packing)**

Le placement des bacs fait référence au placement de plusieurs bacs et peut être trouvé lorsque le matériau de base (contenant) est disponible sous forme de feuilles plutôt que de rouleaux. Le placement de la bande n'est pas limité au boîtier rectangulaire. Dans les applications industrielles, ce problème est appelé découpe de stock, par exemple dans l'industrie du verre, du bois et des métaux.

L'objectif est généralement de trouver l'ensemble des feuilles pour accommoder toutes les parties de la liste de commande en minimisant le total des matériaux utilisés. Selon l'application, les feuilles peuvent être identiques ou avoir des dimensions différentes. Le placement régulier des bacs est fréquent en trois dimensions, notamment les problèmes de chargement des conteneurs et des palettes.

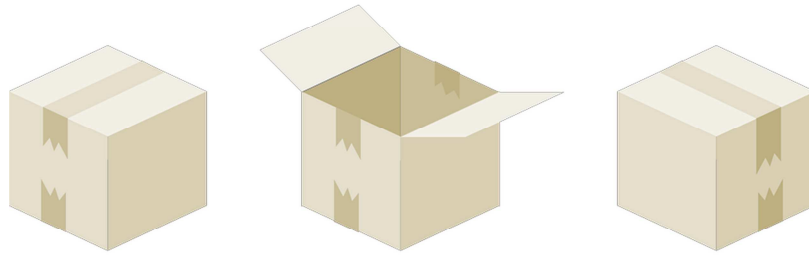


Figure 21 :Le placement en 3D

Disposition de la table non-guillotine

Les applications industrielles impliquant des dispositions non-guillotine sont moins fréquentes. Le problème de découpe de stock 2D se présente comme un problème partiel dans le chargement des palettes et des conteneurs. Une approche couramment adoptée, consiste à réduire le problème 3D à deux dimensions. Une fois qu'un ensemble de boîtes a été déterminé pour une certaine couche, la couche représente un problème d'emballage 2D non-guillotine.

Le problème de placement rectangulaire peut se produire sous la forme non-guillotine lorsque différentes méthodes de coupe telles que le découpage au laser et au plasma sont utilisées pour obtenir les pièces.

10. Conclusion

Dans ce chapitre pour continuer notre concept, nous avons expliqué l'importance des problèmes de placement dans le monde industriel et les différentes contraintes existantes que ce soit dans le placement ou le découpe et on a cité quelque domaine de placement dans le monde industriel, tout cela pour donner une idée de la nécessité de résolution de ce type de problèmes dans le monde concurrentiel.

Chapitre IV

Résolution des problèmes de placement avec des pièces

Rectangulaires

Introduction

Le problème de placement consiste à rechercher le meilleur amalgame au sens des objectifs du placement, et dans le problème de découpe, il s'agit de trouver un ensemble d'amalgames pour satisfaire les demandes. Un amalgame est la manière de découper une unité d'une matière première. Le placement constitue la partie la plus importante du problème de découpe et on ne peut résoudre efficacement un problème de découpe sans résoudre efficacement celui du placement.

Le placement fait partie des problèmes d'optimisation combinatoire qui suscitent beaucoup d'intérêt. Malgré les progrès considérables de l'outil informatique, les méthodes d'énumération, exhaustive ou partielle, sont encore peu satisfaisantes en termes de temps d'exécution ou d'efficacité. Comme ces problèmes contiennent souvent beaucoup de solutions à intérêts pratiques acceptables, les recherches sont orientées vers le développement des méthodes heuristiques. Le but est de trouver une solution de qualité satisfaisante en un temps de calcul raisonnable, d'autant plus que pour des problèmes réels, il n'est pas toujours impératif de trouver la solution optimale, mais des solutions dont la qualité et le temps mis pour les obtenir restent acceptables. [5]

Dans ce chapitre, nous allons nous intéresser à la résolution de ce problème en présentant une heuristique développée récemment puis nous comparerons ses résultats aux résultats obtenus en utilisant un logiciel résolvant ce type de problème.

Partie 1 L'heuristique Least Wasted First(LWF)

1. Introduction

Divers algorithmes heuristiques basés sur différentes stratégies ont été présentés pour résoudre ces problèmes.

On présente d'abord le problème et son modèle mathématique avec une approche de solution qui est basé sur l'algorithme heuristique (a Least Wasted First – LWF) qui évalue les positions utilisées par les rectangles et une recherche locale aléatoire est introduite pour améliorer les résultats, on a aussi posé un processus de packing détaillé pour expliquer les étapes principales de l'heuristique (LWF)

2. Définition du problème

Le problème de placement des pièces rectangulaires fait référence au placement ou la découpe de plusieurs rectangles à partir des feuilles représentant la matière première. Dans les applications industrielles, on peut citer par exemple l'industrie du verre, du bois et des métaux. Généralement il s'agit de trouver l'ensemble des feuilles pour accommoder toutes les parties de la liste de commande en minimisant le total des feuilles utilisées. Selon l'application, les feuilles peuvent être identiques ou avoir des dimensions différentes.

Étant donné une feuille rectangulaire C avec largeur W et hauteur H , un ensemble de n rectangles R avec chaque rectangle R_i de largeur w_i et hauteur h_i ($1 \leq i \leq n$),

Le but du problème de placement de pièces rectangulaires 2D est de trouver un emplacement qui maximise la surface totale des rectangles placé dans la feuille ou le taux de remplissage:

La règle essentiel dans ce placement est d'appliquer le principe du coin inférieur gauche de la feuille BL (Bottom Left = Coin inférieur gauche) à l'origine (0,0) de système de coordonnées cartésiennes et laisse ses quatre côtés parallèlement aux axes X et Y respectivement (figure 25).

Le placement doit satisfaire aux contraintes suivantes

- (1) chaque rectangle placé doit être complètement placé dans la feuille.
- (2) chaque rectangle peut être placé horizontalement ou verticalement dans la feuille, ce qui signifie que les rectangles sont rotatifs.
- (3) chaque bord des rectangles placé doit être parallèle à un bord de la feuille,

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

qui est également appelé placement orthogonal.

(4) les deux rectangles placés ne doivent pas se chevaucher.

Soit f_i ($0 \leq i \leq n$) indique si le rectangle R_i a peut être placé llée dans la feuille ou non, alors $f_i = 1$, sinon $f_i = 0$.

Pour chaque rectangle R_i placé dans la feuille, (x_{li}, y_{li}) indique les coordonnées de son coin inférieur gauche et (x_{ri}, y_{ri}) indique les coordonnées de son coin supérieur droit (figure 25.)

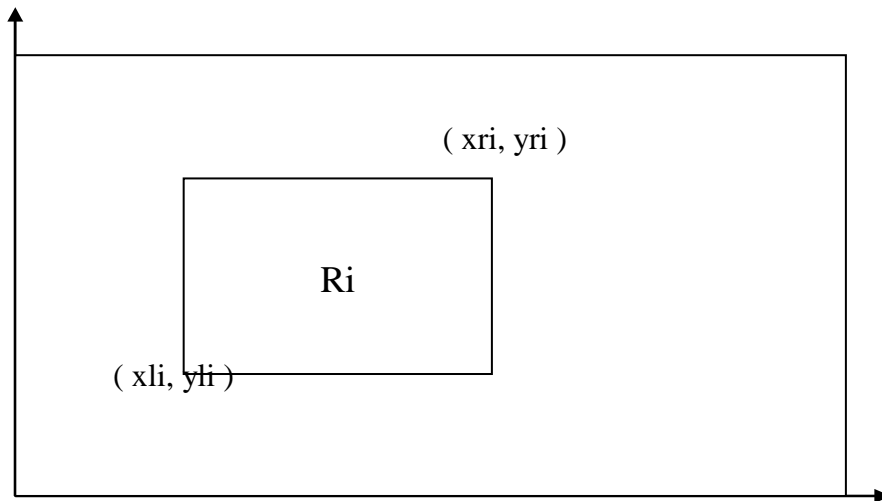


Figure 22. Le rectangle R_i dans le system cartésien

La formulation mathématique du problème

La formulation mathématique du problème peut être décrite comme suit:

$$\max \sum_{i=1}^n f_i w_i h_i$$

Contraintes

$$(1) f_i = 0 \cup (0 \leq x_{li} < x_{ri} \leq w \cap 0 \leq y_{li} < y_{ri} \leq H). \quad i = 1, 2, \dots, n$$

$$(2) f_i = 0 \cup (x_{ri} - x_{li} = w_i \cap y_{ri} - y_{li} = h_i) \cup (x_{ri} - x_{li} = h_i \cap y_{ri} - y_{li} = w_i).$$

$$i = 1, 2, \dots, n$$

$$(3) f_i = 0 \cup f_j = 0 \cup (x_{li} \geq x_{rj} \cup x_{lj} \geq x_{ri} \cup y_{li} \geq y_{rj} \cup y_{lj} \geq y_{ri}). \quad i, j = 1, 2, \dots, i \neq j$$

$$(4) f_i \in \{0, 1\}, i=1, 2, \dots, n$$

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

- (1) Implique tous les rectangles placés sont complètement dans la feuille.
- (2) implique que les rectangles placés sont rotatifs.
- (3) implique que Les rectangles placés ne peuvent pas se chevaucher.
- (4) implique les rectangles peuvent être choisis pour les placer ou non.

3. Résolution du problème

La complexité de ce problème malgré toutes les avancées dans la théorie des mathématiques, on arrive pas a trouver une méthode exacte donnant la solution optimale. Néanmoins plusieurs heuristiques ont été développées donnant des résultats assez satisfaisants du point de vue temps

Parmi ces heuristiques nous allons développer l'Algorithme heuristique Least Wasted First

- **L'heuristique Least Wasted First [3],[5].**

La façon de trouver des positions pour placer les rectangles, qui est présenté par Martello et al.

Le premier rectangle est placé avec son angle **BL** à l'origine.

Soit (**I**) l'ensemble des rectangles placés dans la feuille, les rectangles restants ne peuvent être placés que dans la position **P** sachant que tout rectangle dans **I** soit inférieur à **P** ou à gauche de **P** et ne puisse pas dépasser la bordure de la feuille, plus Formellement, les rectangles à placer peuvent être placés uniquement aux positions contenues dans l'ensemble:

$$S(I) = \{(x, y) : (\exists Ri \in I, x \geq x_{ri} \wedge y \geq y_{ri}) \wedge (x \leq W \wedge y \leq H)\}.$$

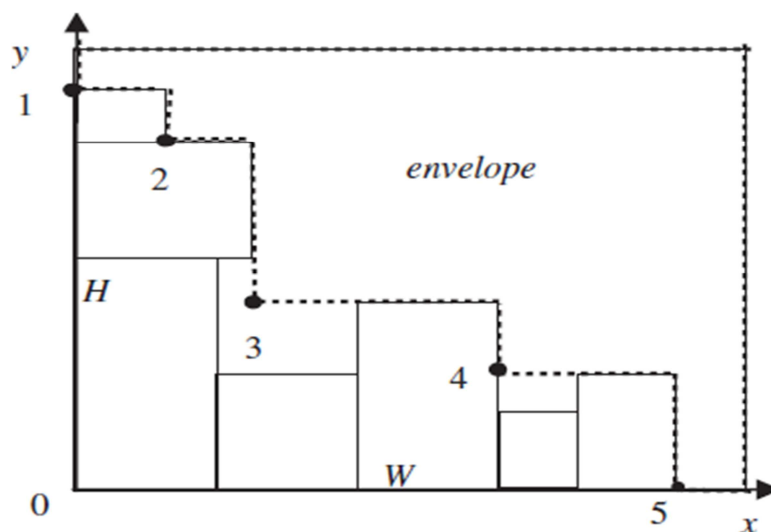


Figure 23. position faisable

Figure 23. montre les régions faisable (la surface entourée par la ligne brisée), les rectangles restants peuvent être placés dans la région appelée **enveloppe**). Il faut noter que nous considérons seulement les points où la pente de l'enveloppe passe de la verticale à l'horizontale (points noirs de la figure 23).

Ces positions sont appelées des positions faisables. Pour vérifier si un rectangle peut être placé à une position possible, il suffit de vérifier si le rectangle dépasse le bord de la feuille ou non.

Il existe une optimisation qui peut être utilisée pour réduire le nombre de positions faisables. Si l'écart entre une position faisable et la bordure de la feuille est inférieur au plus petit bord des rectangles non placés, cette position est appelée **la mauvaise position** (voir la figure 24.) car aucun des rectangles non placés peut être placé à cette position.

Nous pouvons éliminer les mauvaises positions et apporter des modifications à l'enveloppe. Par exemple, dans la (figure.24), si l_1 et l_2 sont plus petits que le plus petit bord des rectangles décompressés, nous éliminons les mauvaises positions 1 et 4, l'enveloppe modifiée est illustrée à la (figure.25) (la région noire est rejetée).

En outre, nous devons considérer quel rectangle et quelle position doit être sélectionnée lorsque le nombre de ces rectangles et positions est supérieur à un.

Nous pouvons voir que chaque position possible est formée par une ligne horizontale et une ligne verticale. Lorsqu'un rectangle est placé dans une position, si sa largeur est égale à la ligne horizontale ou sa hauteur est égale à la ligne verticale, nous appelons ce pack **un bon paquet** (figure 26.)

Nous utilisons une variable de valeur (GN = goodness) pour évaluer ce pack. Si la largeur est égale à la ligne horizontale et que la hauteur est égale à la ligne verticale, GN de ce paquet sera 2;

Si seulement l'un d'eux est égal à la hauteur ou la largeur , GN sera égal à 1;

Sinon GN sera 0.

Nous allons sélectionner le placement avec une plus grande valeur de GN.

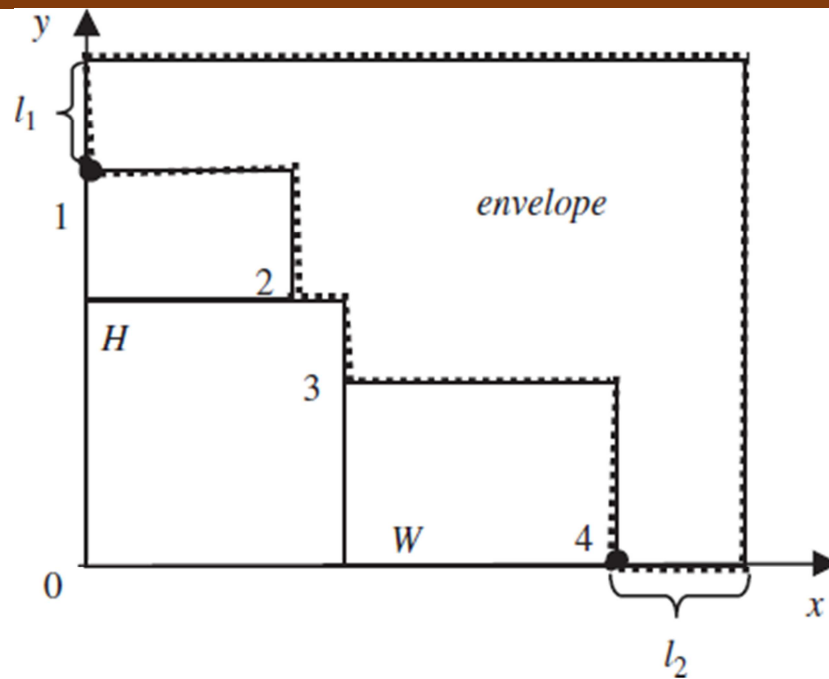


Figure 24. Mauvaises position.

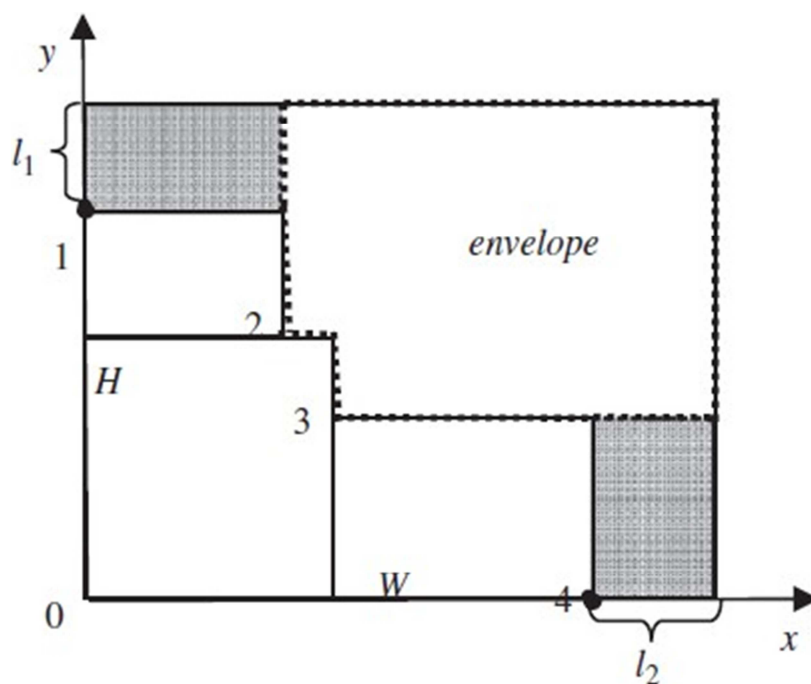


Figure 25. Changer l'enveloppe.

Par conséquent, nous pouvons décrire la manière de sélectionner le rectangle et de positionner comme suit:

(A) sélectionnez le rectangle et positionnez-le de telle sorte que la surface gaspillée soit minimale si le rectangle sélectionné est placé à la position

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

sélectionnée;

(B) si le nombre de ces emplacements est supérieur à un, sélectionnez celui avec la valeur maximale de GN;

(C) si le nombre de ces postes est supérieur à un, sélectionnez la première position (considérez les positions de gauche à droite); et

(D) si le nombre de ces rectangles est supérieur à un, sélectionnez le premier rectangle (considérons les rectangles dans l'ordre donné). En utilisant la stratégie de placement des rectangles décrits ci-dessus, nous obtenons

L'algorithme de base suivant:

2d-packing (R)

$I \leftarrow \emptyset$;

while R n'est pas vide **do**

Calculer l'ensemble des positions faisables P

Utilisez la façon décrite ci-dessus pour sélectionner le rectangle de R et la position P ;

if aucun des rectangles de R ne peut être placé dans l'une des Position en

P **then break** ;

else

nommé le rectangle sélectionné r , ajouté r dans le paquet,

$I \leftarrow I \cup \{r\}$,

$R \leftarrow R \setminus \{r\}$;

end if

end while

return la surface totale des rectangles de I .

end.

Où R est l'ensemble ordonné de rectangles, I est l'ensemble des rectangles placés jusqu'à présent, P désigne l'ensemble de tous les positions faisables.

Dans la procédure 2d-packing (R), l'ensemble I est au départ vide, puis exécuter le processus de placement jusqu'à ce qu'aucun des rectangles ne peut être placé dans la feuille. La surface totale des rectangles placés est retournée à la fin.

Recherche locale aléatoire

Étant donné que le résultat de la procédure 2d-packing (R) dépend de l'ordre des rectangles dans R, nous utiliserons une recherche locale aléatoire pour améliorer la qualité de la solution.

L'algorithme développé est présenté comme suit:

RLs-2d-packing (R, callmax, samemax)

Trier les rectangles dans R par leur surface ;

For $i = 1$ **to** $|R|$ **do**

if $w_i < h_i$ **then**

 Échangez (w_i , h_i);

end if

end for

Meilleur \leftarrow 2d-packing

(R); Même \leftarrow 0;

For $i = 1$ **to** callmax **do**

 Sélectionnez le rectangle a et b de R de manière aléatoire,
 échangez l'ordre de a, b dans R ;

 surf \leftarrow 2d-packing (R);

If surf > Meilleur **then**

 Même \leftarrow 0, la meilleure \leftarrow surf;

if Meilleur = $W * H$ ou Meilleur = totarea **then break**;

end if

 else

 Échangez la position de a, b dans R, même \leftarrow même + 1;

end if

if même > samemax **then**

break;

end if

end for

return le Meilleur

end .

Où :

callmax, **samemax** sont les paramètres utilisés pour contrôler le nombre

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

d'appels de la procédure 2d-packing (R),

meilleur est utilisé pour enregistrer le meilleur résultat trouvé jusqu'ici,

totarea est la superficie totale des rectangles dans **R**.

même est utilisé pour compter le nombre d'appels dans lesquels le résultat n'est pas amélioré.

Dans le rls-2d-packing (**R**, **callmax**, **samemax**) Procédure, nous trions d'abord les rectangles par leur surface puisque les rectangles avec une surface maximale devraient avoir la priorité, puis nous échangeons la hauteur et la largeur des rectangles de sorte que la largeur soit pas plus courte que la hauteur.

Afin d'améliorer le résultat, nous sélectionnons deux rectangles au hasard (la taille des rectangles sélectionnés ne devrait pas être égal), et échanger leur commande en R, appeler 2d-packing (R) pour vérifier si le résultat est amélioré ou non, si oui, nous allons accepter le changement, sinon l'annuler. Si le résultat ne peut pas être amélioré dans les appels **samemax** ou le nombre d'appels dépasse **callmax** ou une solution optimale est trouvée, nous arrêterons le processus.

Algorithme heuristique Least Wasted First

La procédure rls-2d-packing (**R**, **callmax**, **samemax**) est souvent piégée dans les minimums locaux. Inspiré par l'idée de l'algorithme de Las Vegas [3][24], l'algorithme heuristique LWF est comme suit:

Lwf-rls-2d-packing (R, lmax, callmax, samemax)

Meilleur \leftarrow 0;

for i = 1 **to** lmax **do**

surface \leftarrow rls-2d-packing (R, callmax, samemax);

if surface > meilleur **then**

Meilleure \leftarrow surface ;

If meilleur = W * H ou meilleur = totarea

then break; end if

endif end for

return le meilleur

end

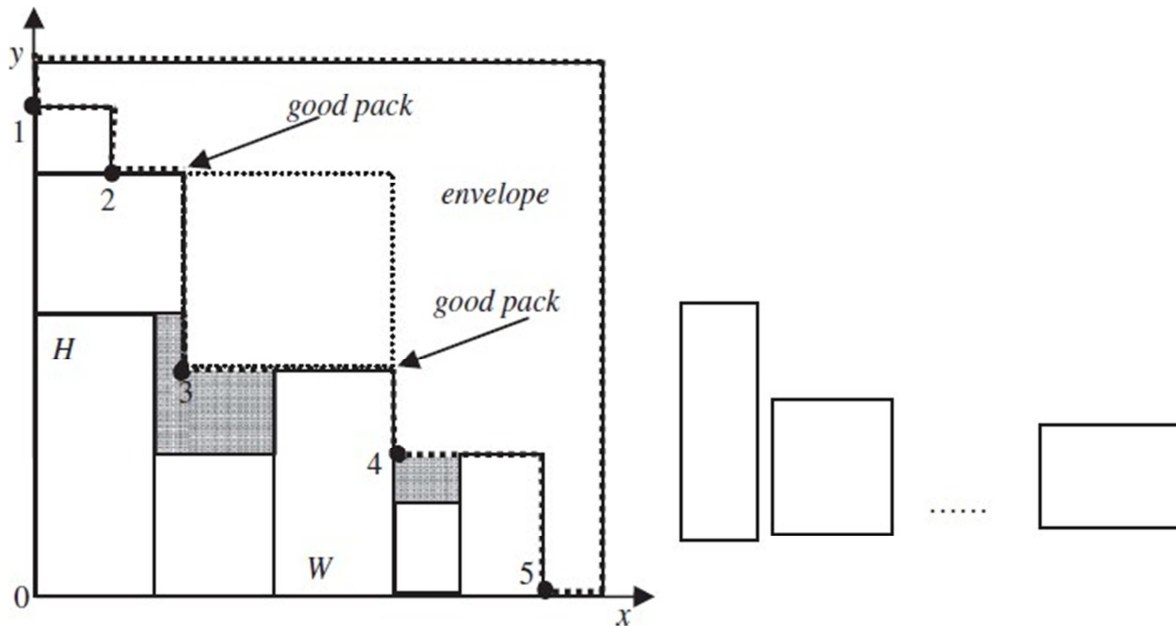


Figure 26. good pack

Où l_{max} est le nombre d'appels de la procédure `rls-2d-packing` (R , `callmax`, `samemax`),

le meilleur est utilisé pour enregistrer le meilleur résultat trouvé jusqu'à présent.

Totarea est la surface totale des rectangles dans R . Dans la procédure `lwf-rls-2d-packing` (R , l_{max} , `callmax`, `samemax`),

nous appelons à plusieurs reprises la procédure `rls-2d-packing` (R , `callmax`, `samemax`) et enregistrez le meilleur résultat trouvé jusqu'à ce que le nombre d'appels dépasse l_{max} ou une solution optimale soit trouvée.

Complexité de l'heuristique LWF

La complexité (temporelle) d'un algorithme est le nombre d'opérations élémentaires (affectations, comparaisons, opérations arithmétiques) effectuées par un algorithme. Ce nombre s'exprime en fonction de la taille n des données. L'heuristique LWF utilise une boucle ou se trouve à l'intérieur un appel à une procédure, cette situation fait que cette heuristique sera de l'ordre exponentiel (dont la complexité est supérieure à tout polynôme en n) que l'on s'accorde à dire impraticables dès que la taille des données est supérieure à quelques dizaines d'unités.

4. Présentation graphique

Le processus de placement étape par étape, selon la restriction ci-dessus :

- La table est placée à l'origine du système de coordonnées et les rectangles sont triés par ordre décroissant selon la taille de la surface.

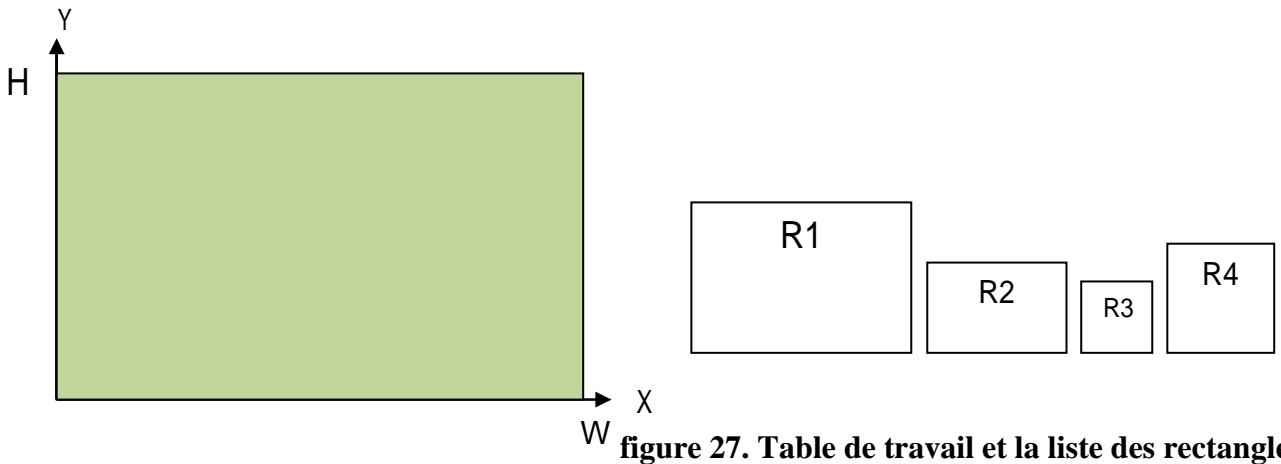


figure 27. Table de travail et la liste des rectangles

Sélection de la première position et placement d'un rectangle :

- Puisque, au début la table d'placement, est vide, la première position libre, la Position 1, est toute la surface de table avec son coin inférieur gauche placé à l'origine.
- Les coordonnées de la position 1 sont des coordonnées de l'origine.
- La valeur calculée pour le facteur de montage pour les quatre rectangles est une, de sorte que celle avec la plus grande taille de la surface est sélectionnée pour être placée, et c'est rectangle R1.
- Le rectangle sélectionné R1 est placé aux coordonnées de la Position 1 sélectionnée (Figure 31)

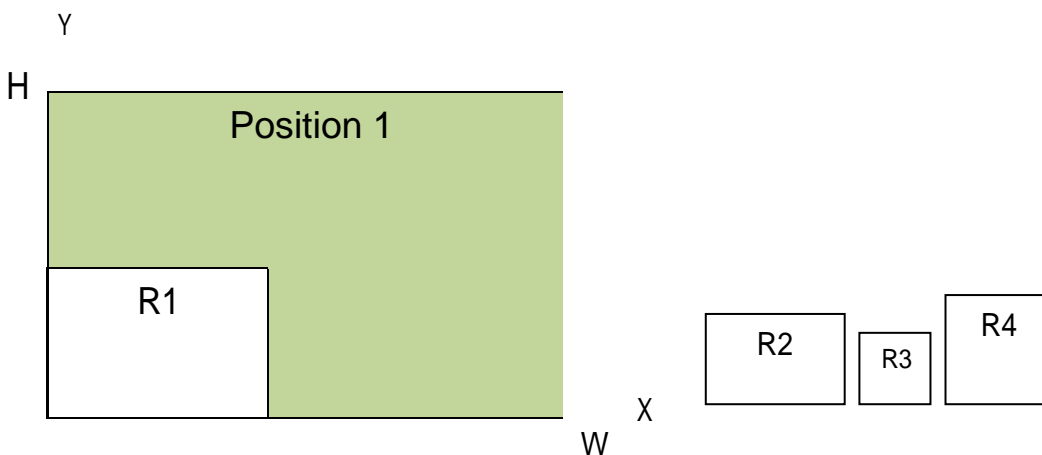


Figure 28.placement du premier rectangle

Trouver de nouvelles positions après le rectangle placée :

- La surface libre sur la table ne répond pas à l'exigence d'une nouvelle position en raison de sa forme et doit être ajustée. En conséquence, à la demande de direction de placement, selon l'axe X de la table, avec l'orientation horizontale exclusivement, la position 1 est divisée en deux formes rectangulaires avec une ligne horizontale depuis le coin supérieur droit du dernier élément placé sur le bord droit de la position 1.
- Maintenant, il existe deux nouvelles positions (Figure 29), la Position 1 et la Position 2, avec ses coordonnées alignées en fonction de l'état des positions de tri. La position précédente 1 est effacée de la liste des positions.
- Les coordonnées de la nouvelle position 1 sont des coordonnées du coin inférieur droit de R1.
- Les coordonnées du nouveau poste 2 sont des coordonnées du coin supérieur gauche de R1.

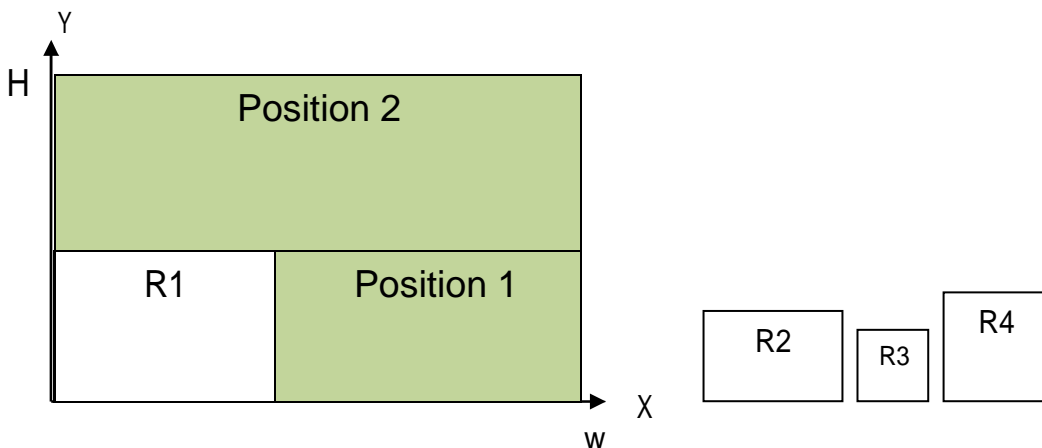


Figure 29. Recherche de position

Placement d'un nouveau rectangle :

- La position sélectionnée est la position 1.
- La valeur calculée pour le facteur de montage pour les trois rectangles est une, de sorte que celle avec la plus grande taille de la surface est sélectionnée pour être placée, et c'est le rectangle R4.
- Le rectangle sélectionné R4 est placé aux coordonnées de la Position 1 sélectionnée.

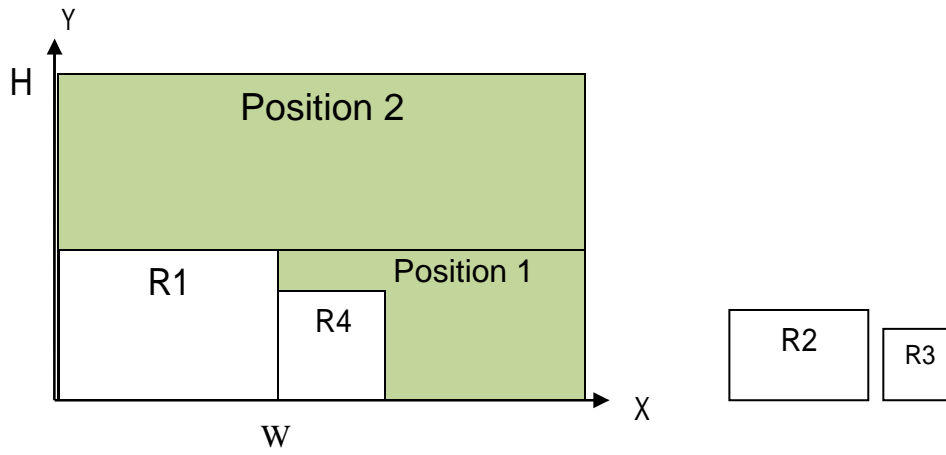


Figure .30 placement de deuxième rectangle

Trouver de nouvelles positions :

- La situation sur la table est similaire à la situation après avoir placé le premier rectangle de sorte que les étapes sont également similaires. Diviser la position 1 en deux formes rectangulaires avec une ligne horizontale du coin supérieur droit du dernier élément placé sur le bord droit de la position 1.
- Il existe maintenant deux nouvelles positions et un ancien (Figure .31), avec ses coordonnées alignées en fonction de la condition pour les positions de tri.
- Position 1 avec coordonnées dans le coin inférieur droit de R4.
- Position 2 avec des coordonnées dans le coin supérieur gauche de R4.
- Position 3, ancienne position 2, avec des coordonnées dans le coin supérieur gauche de R1.

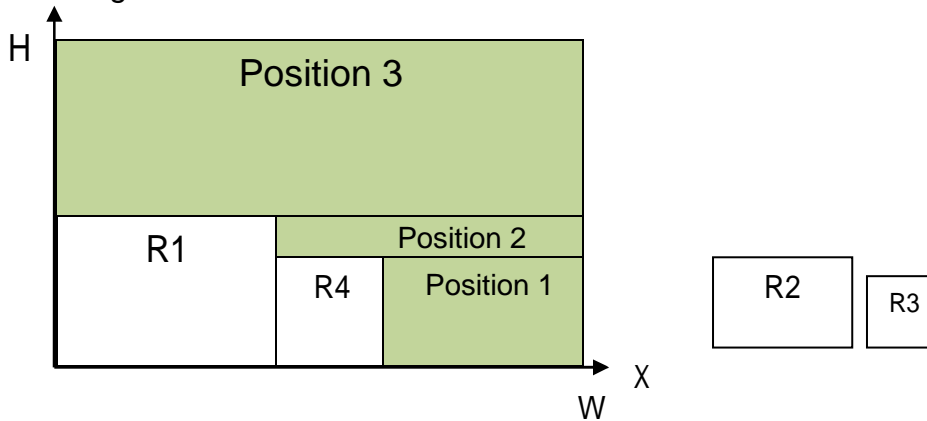


Figure .31 Recherche d'une nouvelle position

Cas de position inadéquat

- La position 2 est insuffisante en raison de sa largeur et doit être rejetée. Il ne reste plus aucun rectangle pouvant correspondre à la largeur de la position 2.
- Après avoir effacé la position 2, la position 1 doit être réglée comme indiqué ci-dessous (Figure .32)
- Coloré en blanc fait partie de la Position 2 rejetée et de l'espace de table inutilisé.
- Position 3 est maintenant position 2.

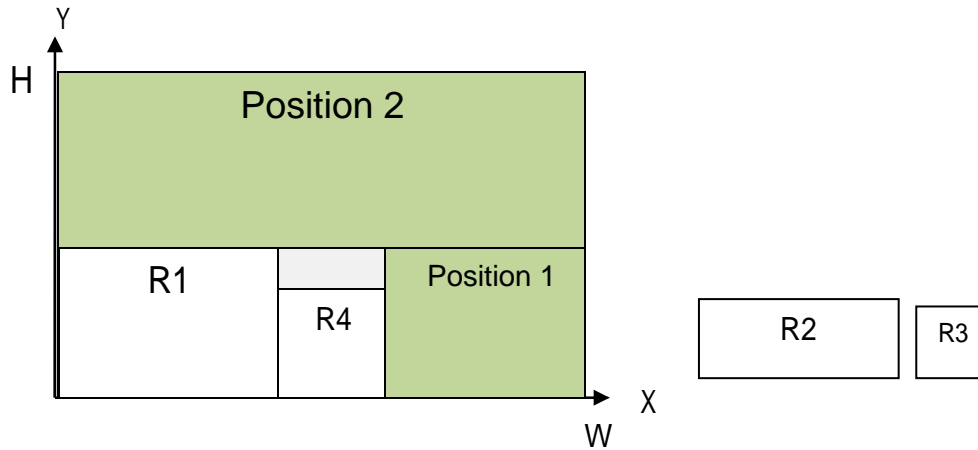


Figure .32 éliminé la mauvaise position

- Le processus se poursuit en emballant les deux derniers rectangles et en trouvant de nouvelles positions.

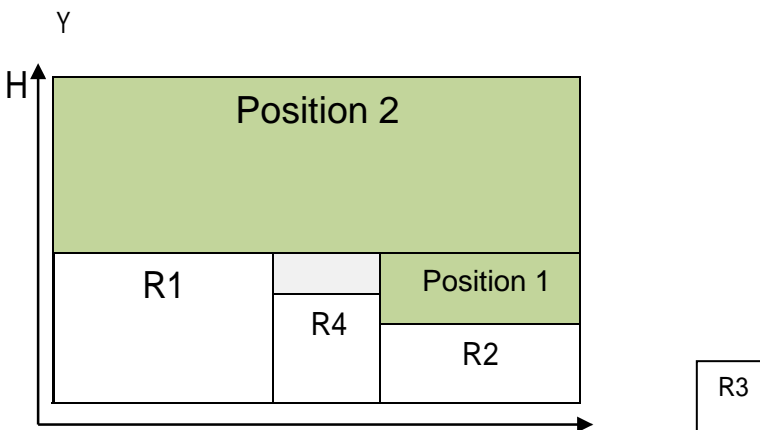


Figure .34 placement du 3eme rectangle

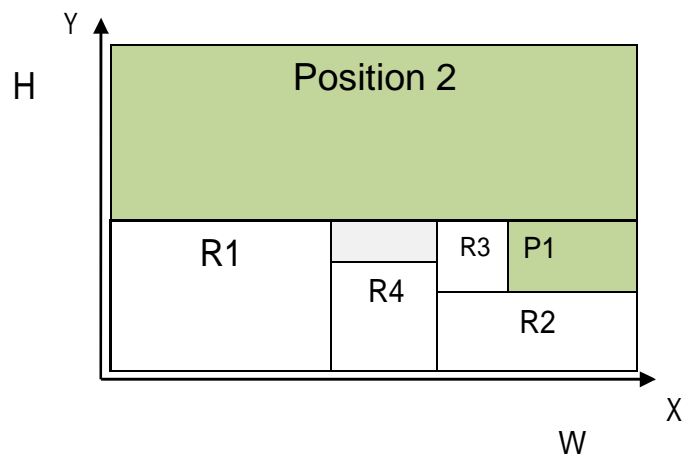


Figure 35. placement le 4eme rectangle

Le processus s'est terminé car il n'y a plus rectangle libre. Tous les rectangles sont placés.

En fin, les rectangles placés sur la table (Figure 36).

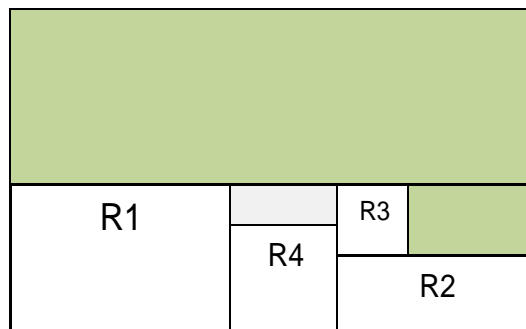


Figure .36.plan final

Partie 2 : Logiciel OptiCoupe V

1. Introduction

Les problèmes d'optimisation de placement de rectangles font généralement intervenir un nombre très élevé de combinaisons, qui ne peuvent être toutes examinées du fait de la durée des calculs.

Par conséquent, un logiciel d'optimisation ne pourra pas identifier "la meilleure solution" d'une manière systématique (*cette solution pouvant d'ailleurs ne pas exister, ou ne pas être unique, selon les critères*), mais il s'en approchera d'autant plus que les critères de sélection des cas examinés, par opposition aux cas éliminés, seront choisis avec pertinence.

Il existe plusieurs logiciels permettant de résoudre les problèmes de placement de pièces rectangulaires. Nous avons opté pour le Logiciel OptiCoupe V pour résoudre notre application.

1.1. Présentation

OptiCoupe V est un logiciel d'optimisation de placement de rectangles et même de barres, développé et édité par **BOOLE & PARTNERS**, et qui bénéficie de l'expérience de plusieurs milliers d'utilisateurs situés dans plus de 30 pays.

Il est principalement utilisé dans les secteurs du bois, du métal, du plastique, du verre, et de la pierre, et trouve son application dans les métiers du négoce de matériaux (*panneaux, tôles, vitrages, profilés et poutrelles*), de la menuiserie générale (*bois ou métal*), de l'agencement, de la façade, de la charpente, etc.

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

1.2. Installation

OptiCoupe V peut être installé sur tout ordinateur de type PC fonctionnant sous Windows (*Windows 98 / 2000 / XP Pro / Vista*), à partir du fichier "**InstallOptiCoupe.exe**" disponible sur le CD du logiciel.

L'exécution du fichier "**InstallOptiCoupe.exe**" affiche la fenêtre suivante :



2. Principes généraux

2.1. Structure générale des stocks

La structure des stocks d'**OptiCoupe V** comporte deux parties :

- Une **liste de matériaux** contenant les désignations des matériaux identifiés dans **OptiCoupe V**, et utilisables dans les listes de rectangles,
- Une **liste de formats** contenant l'ensemble des formats disponibles en stock pour chacun des matériaux de la liste précédente.

Par exemple, si l'on souhaite créer une feuille de **3050 x 1850** en mélaminé blanc de **19 mm**, il conviendra de procéder en deux étapes :

- 1- Création d'un matériau de panneau **MELA_BL** d'épaisseur **19 mm**,
- 2- Création d'un panneau de **3050 x 1850** rattaché au matériau **MELA_BL, 19**.

Si le matériau **MELA_BL, 19** est disponible dans plusieurs formats, il suffit de les créer successivement, et de les rattacher à ce même matériau.

Si l'on utilise le mélaminé blanc dans une autre épaisseur, par exemple **16 mm**, il s'agit alors d'un autre matériau, qu'il convient de créer et d'utiliser distinctement du précédent.

*Les stocks d'**OptiCoupe V** peuvent être gérés soit avec les quantités réelles disponibles, soit avec des quantités illimitées, lorsque les commandes sont passées en fonction des besoins.*

2.2. Fonctions multi-matériaux et multi-formats

OptiCoupe V est un logiciel **multi-matériaux** et **multi-formats**.

La fonction **multi- matériaux** permet de saisir ou d'importer des listes de rectangles comportant plusieurs matériaux dans un ordre quelconque.

Préalablement à l'optimisation, cette fonction regroupe les lignes de rectangles de même matériau, puis lance un calcul d'optimisation distinct pour chacun des matériaux de la liste.


Au cours de l'optimisation, la fonction **multi-formats** examine tous les formats disponibles du matériau courant, et retient, plan de coupe par plan de coupe, le format le plus approprié, qui peut être soit un module entier, soit une chute issue d'un précédent débit.

Ainsi le résultat d'une optimisation comporte une combinaison des formats disponibles.

2.3. Recherche du format optimum

Cette fonction, permet de rechercher automatiquement le format de feuille ou de barre unique le plus approprié à un rectangle donné.

La recherche s'effectue sur la base d'un stock virtuel, indépendamment du stock effectif qui existerait pour le matériau concerné.

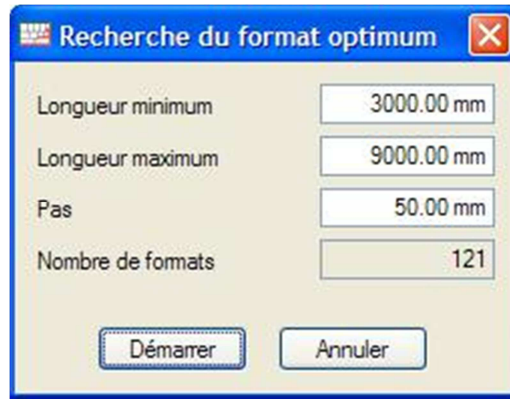


Hauteur minimum	1500.00 mm	Largeur minimum	1000.00 mm
Hauteur maximum	1500.00 mm	Largeur maximum	4500.00 mm
Pas	50.00 mm		
Nombre de formats	71	Démarrer	Annuler

Dans le cas ci-dessus, **OptiCoupe V** testera les 71 formats virtuels, et suggérera par exemple d'utiliser le format 1500 x 3850, s'il est mieux adapté que les 70 autres formats.

Ce résultat pourra être trouvé même si aucun format de panneau n'est disponible en stock.

Pour les débits de barres, la démarche est similaire :



Dans le cas ci-dessus, **OptiCoupe V** testera les 121 formats virtuels, et aboutira par exemple à une solution en 17 barres d'une même longueur de 8250 mm.

La fonction de recherche du format optimum est particulièrement intéressante lorsqu'il est possible de commander ses formats de feuilles ou de barres sur mesure.

A noter toutefois que cette fonction nécessite une durée de calcul d'autant plus importante que le nombre de formats testés est élevé.

3. Saisie des données

3.1. Matériaux et stocks de feuilles

Les deux commandes "Matériaux des panneaux" et "Edition des panneaux" du menu "Stock"

ouvrent les fenêtres ci-dessous :

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

Édition des matériaux des panneaux

Matériau	Description	Épaisseur	Fil	Coût de la découpe linéaire	Prix (€/m ²)
PLW_305		19.00	Non	1.00	15.00
MELA_BL		19.00	Non	1.00	10.00
MDF_280		20.00	Non	1.00	12.00

Matériau: Épaisseur (mm): Fil

Description: Prix (€/m²):

Coût de la découpe linéaire (€/m):

Édition du stock de panneaux

Affichage des matériaux:

Hauteur	Largeur	Référence	Matériau	Quantité	Réservé	Affr. H.	Affr. L.	Prix	Surface
3050.00	2030.00		PLW_305, 19.00	Illimité	Non	0.00	0.00	15.00	6.19 m ²
3050.00	1850.00		MELA_BL, 19.00	Illimité	Non	0.00	0.00	10.00	5.64 m ²
2800.00	2070.00		MELA_BL, 19.00	Illimité	Non	0.00	0.00	10.00	5.80 m ²
2800.00	2070.00		MDF_280, 20.00	Illimité	Non	0.00	0.00	12.00	5.80 m ²

Hauteur (mm): Largeur (mm): Référence:

Matériau: Prix (€/m²): Réserve Illimité

Chute:

Observations:

Affranchissements spécifiques: Hauteur (mm): Largeur (mm):

Les affranchissements spécifiques à chaque feuille sont indépendants des affranchissements généraux, auxquels ils peuvent se substituer ou se rajouter, selon les options choisies.

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

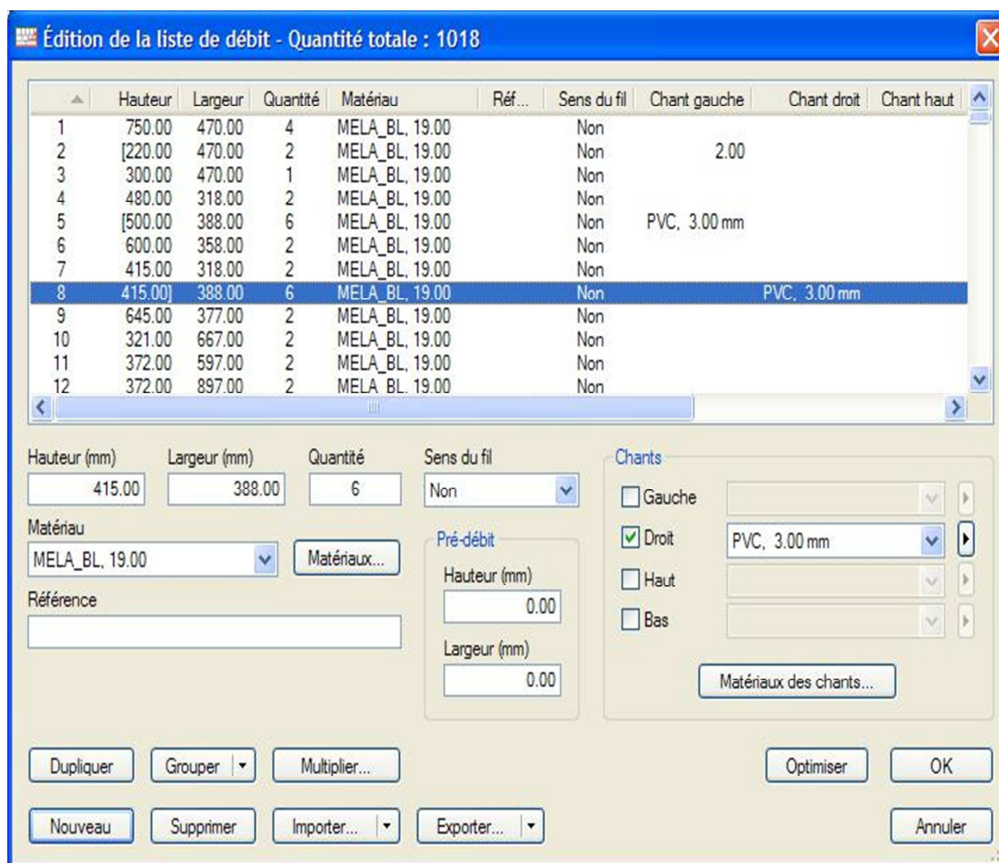
3.2. Liste des rectangles

La commande "Nouveau débit de panneaux" du menu "Fichier" ouvre la fenêtre ci-dessous, qui permet de renseigner la liste de débit soit par saisie directe, soit par import de fichier.

A chaque pièce peut être associée une référence principale, et jusqu'à 16 références Complémentaires, à créer et à paramétrer dans la fenêtre "Outils / Préférences".

Chacun des 4 chants d'une pièce peut-être défini soit par son matériau, soit par son épaisseur.

Les épaisseurs des chants sont déduites des dimensions finies (*saisies*), tandis que les valeurs de pré-débit leur sont rajoutées, pour obtenir les dimensions nettes de découpe.



Les touches "MAJ" et "CTRL" du clavier permettent de sélectionner simultanément plusieurs lignes de la liste de débit (*fonction de multi-sélection*).

Le bouton "Dupliquer" duplique les lignes de débit sélectionnées.

Le bouton "Grouper" regroupe les lignes dont les contenus sont identiques en une seule ligne.

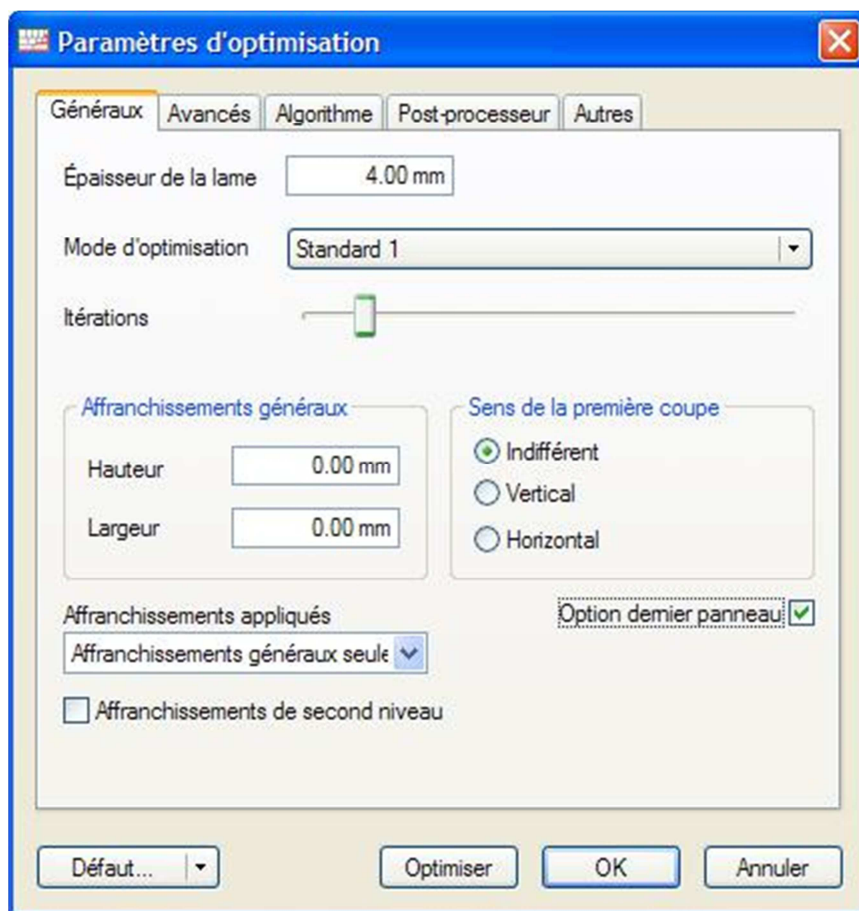
Le bouton "Multiplier" multiplie les quantités des lignes sélectionnées par un même nombre, et s'applique à la totalité de la liste en l'absence de sélection.

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

3.3. Paramètres d'optimisation des feuilles

Paramètres généraux

L'onglet "Généraux" de la fenêtre "Paramètres d'optimisation" se présente comme suit :



Dans un souci de simplification, **OptiCoupe V** propose plusieurs modes d'optimisation pré-programmés, ayant des paramètres algorithmiques prédéfinis.

Par rapport aux modes "Rapide" et "Standard", les modes "Avancé" autorisent le changement de sens de la découpe principale (*retournement*), permettant d'obtenir de meilleurs rendements et des chutes mieux réutilisables, mais au prix de plans de coupe souvent plus complexes.

Les modes "Commande Numérique" sont spécifiques aux contraintes des scies numériques, et dépendent des gammes des différents constructeurs.

Le curseur "Itération" gère le brassage statistique du débit, et permet d'améliorer les résultats de l'optimisation tout en respectant les contraintes imposées.

Les meilleures solutions sont généralement trouvées avant la 10^{ème} itération, mais dans certains cas, elles pourront apparaître après la 50^{ème} itération.

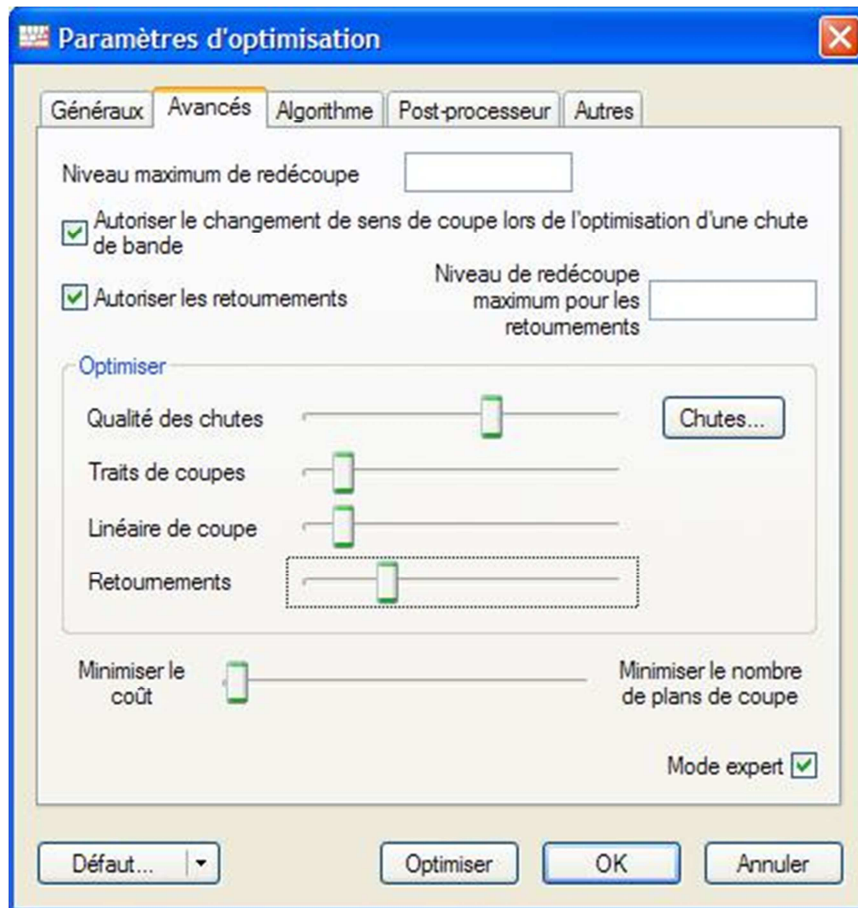
Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

L'option "Affranchissement de second niveau" applique un nouvel affranchissement au cours de la découpe, aux niveaux de redécoupe supérieurs à 2.

L'option "Dernier panneau" permet, lorsqu'un matériau est disponible dans plusieurs formats, de retenir en priorité le format qui solde le débit en plaçant toutes les dernières pièces.

Paramètres avancés

L'onglet "Avancés" donne accès aux paramètres suivants :



Les niveaux maximum de redécoupe permettent de limiter le nombre de combinaisons à tester et de ce fait, limitent également la complexité des plans de coupe et la durée des calculs.

Les 4 curseurs du cadre "Optimiser" permettent, une fois les pièces du plan de coupe définies, de choisir le schéma de découpe le plus proche des préférences choisies.

Plus le curseur est à droite, et plus le critère intervient dans la pondération des préférences.

Le dernier curseur permet de rechercher des solutions comportant moins de plans de coupe, au détriment d'une augmentation raisonnable du taux de chute.

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

4. Édition des résultats

4.1. Édition des optimisations de feuilles

La fonction d'édition imprime une synthèse de l'optimisation, suivie par les plans de coupe.

OptiCoupe 5.12c OptiCut Page 1 / 3	BOOLE & PARTNERS	21/08/2009
--	-----------------------------	------------

Liste de débit

	Matériau	Référence	Dimension	Quantité
1	MELA_BL, 19.00	Ref_D	750.00 × 470.00	4
2	MELA_BL, 19.00	Ref_E	220.00 × 480.00	2
3	MELA_BL, 19.00		300.00 × 640.00	1
4	MELA_BL, 19.00	Ref_A	506.00 × 318.00	4
5	MELA_BL, 19.00		505.00 × 425.00	10
6	MELA_BL, 19.00	Ref_C	600.00 × 372.00	2
7	MELA_BL, 19.00	Ref_F	415.00 × 566.00	5
8	MELA_BL, 19.00		645.00 × 377.00	2
9	MELA_BL, 19.00	Ref_G	321.00 × 720.00	2
10	MELA_BL, 19.00		372.00 × 597.00	6
11	MELA_BL, 19.00	Ref_H	372.00 × 897.00	2
12	MELA_BL, 19.00	Ref_B	2045.00 × 355.00	2
TOTAL				42

Panneaux utilisés

Matériau	Dimension	Quantité	Surface
MELA_BL, 19.00	3050.00 × 1850.00	2	11.29 m ²

Liste des plans de coupe

	Matériau	Dimension	Quantité	Pièces	Taux de chutes	Coût net
1	MELA_BL, 19.00	3050.00 × 1850.00	1	22	4.51 %	90,04 €
2	MELA_BL, 19.00	3050.00 × 1850.00	1	20	7.20 %	86,82 €
TOTAL			2	42	5.85 %	176,86 €

Récapitulatif

Données techniques		Coûts	
Nombre de panneaux utilisés	2	Coût des pièces	106,24 €
Nombre de plans de coupe	2	Coût en panneaux	112,85 €
Surface totale des panneaux	11.29 m ²	Coût en chutes	6,61 €
Surface totale des pièces	10.62 m ²	Coût des chutes non réutilisables	6,61 €
Taux de chutes	5.85 %	Coût du linéaire de découpe	64,01 €
Taux des chutes non réutilisables	5.85 %	Coût net total	176,86 €
Linéaire de découpe	64.01 m		
Linéaire des chants	0.00 m		

Le menu "Fichier / Configuration de l'impression" permet de personnaliser l'en-tête de la page de garde, et de paramétrer l'édition en sélectionnant les sections à imprimer, les deux couleurs des chutes (*récupérables et non récupérables*), les marges d'impression, et le mode d'édition des plans de coupe (*pleine page, ou demi- page avec détails*).

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

De même, lorsque l'optimisation conduit à l'échec du placement d'une ou de plusieurs pièces, il se crée une section intitulée "Pièces non placées" qui liste chacune des pièces concernées, et les motifs de l'échec correspondant (*stock de feuille épuisé, feuilles trop petites, etc.*).

Édition des plans de coupe de panneaux

Les plans de coupe des panneaux sont imprimés sur la base d'un seul plan par page A4. chaque plan indique le nom du fichier de débit, la date courante, le numéro de page et celui du plan de coupe, le nom du matériau, et les dimensions du panneau à débiter.

OptiCoupe 5.12c OptiCut Page 2 / 3	BOOLE & PARTNERS	21/08/2009
--	-----------------------------	------------

12 – MEA_EL, 19.00 – 3050.00 × 1850.00 – Exemple unique

372.00 × 897.00 Ref_H		372.00 × 600.00 Ref_C	720.00 × 321.00 Ref_G
372.00 × 897.00 Ref_H		372.00 × 600.00 Ref_C	
750.00 × 470.00 Ref_D	566.00 × 415.00 Ref_F	372.00 × 597.00	720.00 × 321.00 Ref_G
		372.00 × 597.00	
750.00 × 470.00 Ref_D	566.00 × 415.00 Ref_F	372.00 × 597.00	506.00 × 318.00 Ref_A
		372.00 × 597.00	506.00 × 318.00 Ref_A
750.00 × 470.00 Ref_D	566.00 × 415.00 Ref_F	372.00 × 597.00	506.00 × 318.00 Ref_A
		372.00 × 597.00	

Les plans sont édités dans l'ordre dans lequel ils ont été trouvés par l'algorithme, après un tri sur le sens de la première coupe (*horizontal, puis vertical*).

Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

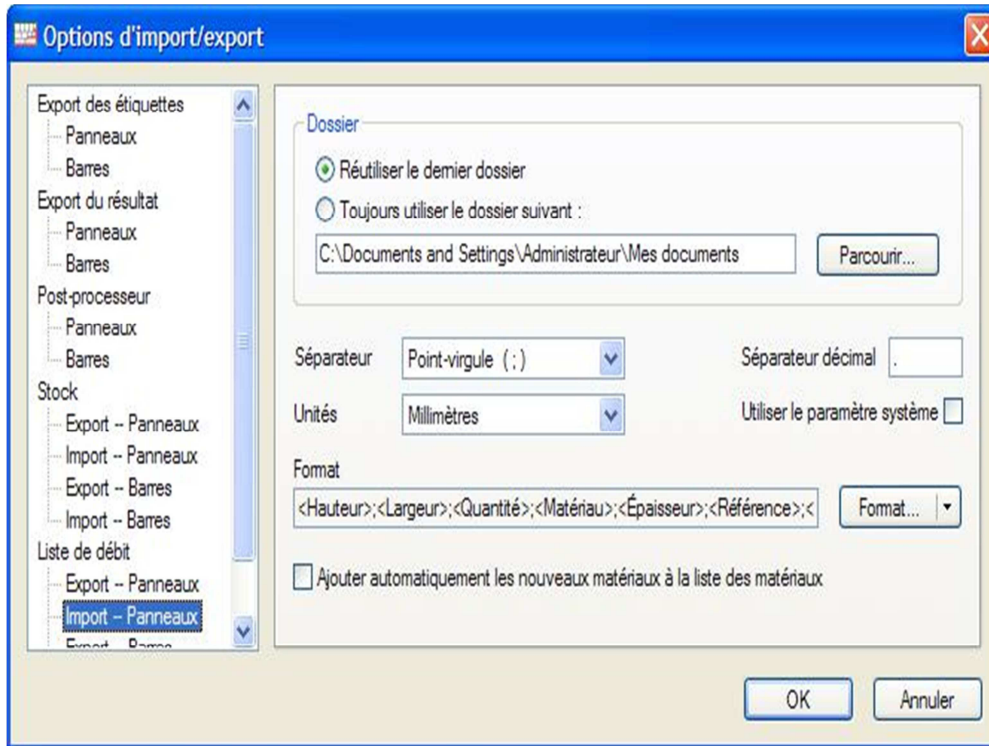
Lorsque le plan de coupe est édité en mode "Demi-page", la moitié inférieure de la page est utilisée pour imprimer soit la liste des pièces (*individuelle ou groupée*), soit la liste des traits de coupe en fonction des options de l'onglet "Panneaux" du menu "Outils / Préférences".

Lorsqu'un même plan de coupe doit être réalisé N fois, la mention "Exemplaire unique" est remplacée par "A fabriquer en N exemplaires".

5. Imports et exports

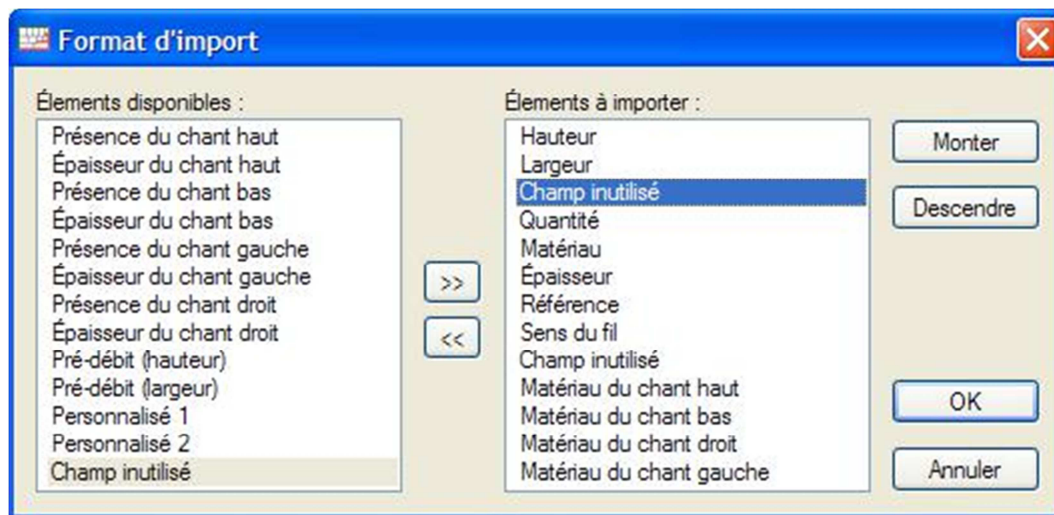
5.1. Import des listes de débits et des stocks

OptiCoupe V permet d'importer les listes de débits et les stocks de panneaux et de barres dans des formats "Texte" paramétrables (*Excel CSV, etc.*).



Chapitre 4 Résolution des Problèmes de placement avec des pièces rectangulaires

Le bouton "Format" ouvre la fenêtre ci-après, permettant de définir les rubriques à importer, et l'ordre dans lequel elles se trouvent dans le fichier texte.



La rubrique "Champ inutilisé" est utilisable plusieurs fois, et permet d'éviter d'importer certaines données du fichier texte qui n'ont pas de correspondant dans **OptiCoupe V**.

4.2. Export des données

En complément aux listes de rectangles et aux stocks, qu'il exporte dans un format texte distinct du format d'import, **OptiCoupe V** exporte un fichier "Étiquettes" et un fichier "Résultats".

Fichier Étiquettes

Le fichier "Étiquettes" comporte, pour chaque pièce produite, chute récupérable, ou feuille utilisée, une ligne dont le contenu est paramétrable.

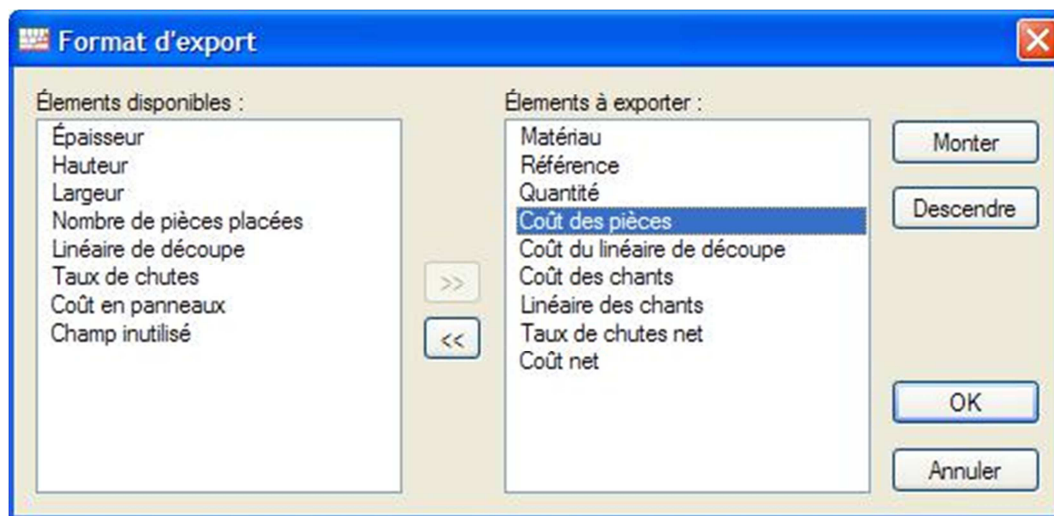
La liste des rubriques de ce fichier est proche de celle de la fonction d'impression d'étiquettes, avec cependant quelques différences.

En particulier, le type d'étiquette (Panneau, Pièce, ou Chute) est défini par un texte éditable.

Fichier Résultats

Le fichier "Résultats" comporte une ligne par plan de coupe.

Chaque ligne contient les données quantitatives permettant de chiffrer le coût net des pièces produites dans le plan de coupe, en tenant compte de la découpe et des chants plaqués.



L'exploitation du fichier "Résultats" par un tableur, ou en liaison avec un logiciel de gestion, permet de définir le besoin total réparti par type de panneau, ainsi que les coûts associés.

Ligne de commande Windows

OptiCoupe V peut être lancé à partir d'un autre logiciel par une ligne de commande Windows qui importera le débit et exportera le fichier "Résultats" en fin d'optimisation.

Ce fichier pourra alors être importé en toute transparence par le logiciel qui intègre cette ligne de commande, en vue de planifier la production et les approvisionnements.

*La syntaxe de cette ligne de commande est disponible auprès de **BOOLE & PARTNERS**.*

Chapitre V

Application pratique

Chapitre 5 Application pratique

1. Introduction

Dans ce chapitre, consacré à la réalisation et la mise œuvre de notre application placement des pièces rectangulaire, l'étude pratique présentée a été menée dans une entreprise de meubles. Le problème posé était le suivant : comment découper des étagères de divers formats dans de grands panneaux de bois de dimensions données, de façon à obtenir, une chute minimale de la surface du matériau utilisé?

Ce problème classique a donné lieu à de nombreux développements théoriques mais nous avons voulu insister ici sur les aspects concrets inhérents à toute étude en milieu industriel. Nous verrons par exemple que, bien qu'il s'agisse d'un problème de découpes à deux dimensions, l'existence de contraintes techniques très fortes réduit considérablement l'aspect combinatoire du problème. Cela nous a permis, après avoir utilisé les deux méthodes développées au chapitre 4 de se familiariser avec les problèmes pratiques.

2. Application

La société « Surfaces » vend des bibliothèques en « Kit » (éléments séparés); elle commande les étagères à l'entreprise M qui les découpe dans des panneaux de bois de largeur et longueur données et constantes ($H \times W$).

Les chutes lui étant facturées, c'est « Surfaces » qui doit indiquer les découpes à effectuer, tout en tenant compte des contraintes techniques importantes imposées par la machine perfectionnée qu'utilise M.

L'entreprise M dispose d'une machine qui comporte une série de scies découpant les panneaux longitudinalement, et deux séries de scies transversales. Une pile de panneaux est placée sur la machine qui les fait alors glisser vers les scies. Aucune manipulation extérieure n'intervient avant la fin des découpes qui sont effectuées de bout en bout (découpe guillotine).

Chapitre 5 Application pratique

Le problème qui nous a été proposé par M est le suivant :

On dispose d'une suite de 10 rectangles (étagères) de dimension hi et $wi, 1 < i < 10$ donnée dans le tableau suivant :

Etagères	Hi	Wi
r1	500	200
r2	400	100
r3	500	200
r4	200	700
r5	800	500
r6	200	500
r7	400	200
r8	300	200
r9	800	100
r10	200	700

à découper dans une feuille (panneau) de dimension $H = 1000$ et $W = 500$

L'objectif est de minimiser le nombre de feuille à utiliser et de réduire les chutes.

Nous présentons dans notre travail les deux méthode développées au chapitre 4, pour déterminer ces découpes de façon à satisfaire aux besoins de « Surfaces » tout en ayant un niveau de chute « acceptable »

3. Résolution par l'heuristique LWF

On appliquera l'heuristique LWF pour placer ces rectangles dans un nombre minimal de feuille.

On doit au préalable réarranger les rectangles de façon à avoir $hi > wi$

Rectangle	Hi	wi
r1	500	200
r2	400	100
r3	500	200
r4	700	200
r5	800	500
r6	500	200
r7	400	200
r8	300	200
r9	800	100
r10	700	200

1 étape :

➤ Calcule des positions faisables

P	OX	OY	H	W	S
	0	0	1000	500	500000

Au début la feuille est vide on a une seule position de dimension H=1000 et W=500 Dont l'origine est le coin inférieur gauche.

❖ On calcule la surface, la faisabilité et le facteur de montage G_n

$$Si = hi * wi$$

$$Fi = \begin{cases} 1 \\ 0 \end{cases}$$

si $hi < H$ et $wi < W$
sinon il déborde des bordures de la feuille

$$Gn = \begin{cases} 2 & \text{si } hi = (H \text{ ou } W) \text{ et } wi = (W \text{ ou } H) \\ 1 & \text{si } hi = (H \text{ ou } W) \text{ ou } wi = (W \text{ ou } H) \\ 0 & \text{sinon on précèdera à retourner le rectangle} \end{cases}$$

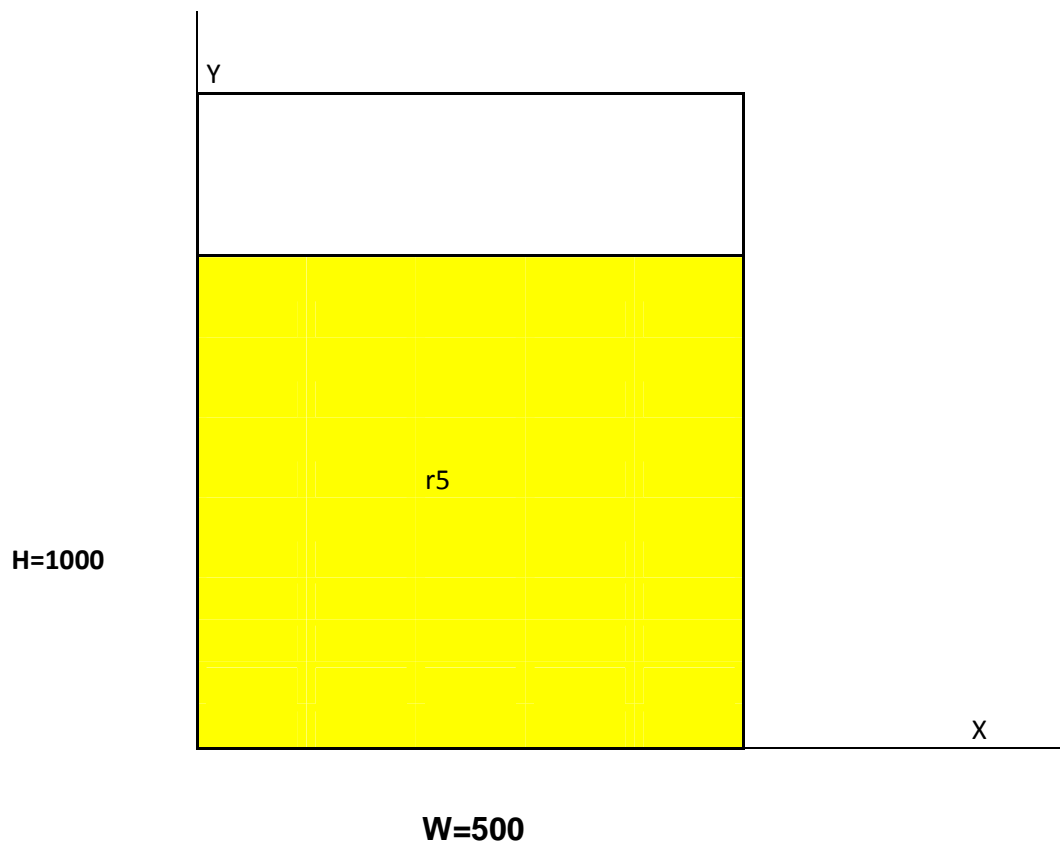
Rectangle	Hi	Wi	Si	Fi	GN/ P
r1	500	200	100000	1	1
r2	400	100	40000	1	0
r3	500	200	100000	1	1
r4	700	200	140000	1	0
r5	800	500	400000	1	1
r6	500	200	100000	1	1
r7	400	200	80000	1	0
r8	300	200	60000	1	0
r9	800	100	80000	1	0
r10	700	200	140000	1	0

- Quel rectangle choisir ?

On élimine tous les rectangles dont F_i sont nuls

On trie par ordre croissant de Gn_i puis on cas d'égalité on prendra celui qui a la plus grande surface puis on cas d'égalité on prendra celui qui as le plus grand h.

Le rectangle candidat à être placer et le rectangle r5 dont $Gn=1$ et surface est 400000



L'ensemble R des rectangles non placées sera réduit de r5

L'ensemble I des rectangles placés sera augmenté de r5

Etape 2 :

1-calcul des positions faisables :

P	OX	OY	H	W	S
	0	0	1000	500	500000

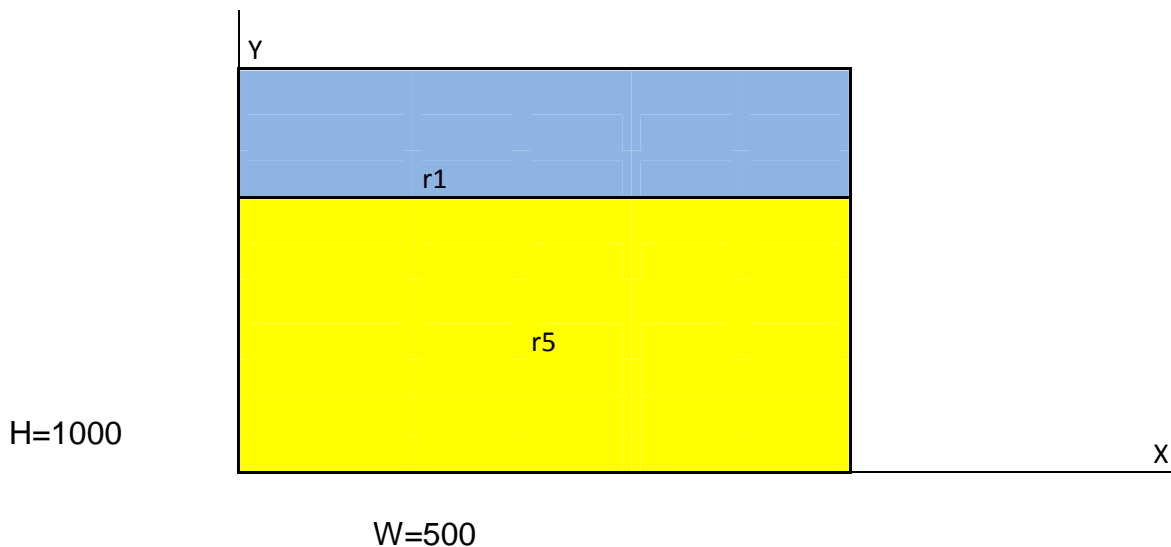
En plaçant r5 (800,500) une nouvelle position est crée

P1	OX	OY	H	W	S
	0	800	200	500	100000

Rectangle	hi	wi	Si	Fi	GN/ P
r1	500	200	100000	1	2
r2	400	100	40000	1	0
r3	500	200	100000	1	2
r4	700	200	140000	0	
r6	500	200	10000	1	2
r7	400	200	80000	1	1
r8	300	200	60000	1	1
r9	800	100	80000	0	
r10	700	200	140000	0	

On prendra r1 on lui faisant une rotation

On a trois rectangle candidats équivalents on prend un des trois au hasard



L'ensemble R= {r2,r3,r4,r6,r7,r8,r9,r10}

L'ensemble I= {r5,r1}

Etape 3 :

Chapitre 5 Application pratique

1-calculs des positions faisables :

P1	OX	OY	H	W	S
	0	800	200	500	100000

En plaçant r1 (500,200)

P2	OX	OY	H	W	S
	0	1000	0	0	0

La feuille est pleine, on procédera a l'ouverture d'une nouvelle feuille

Etape 3 :

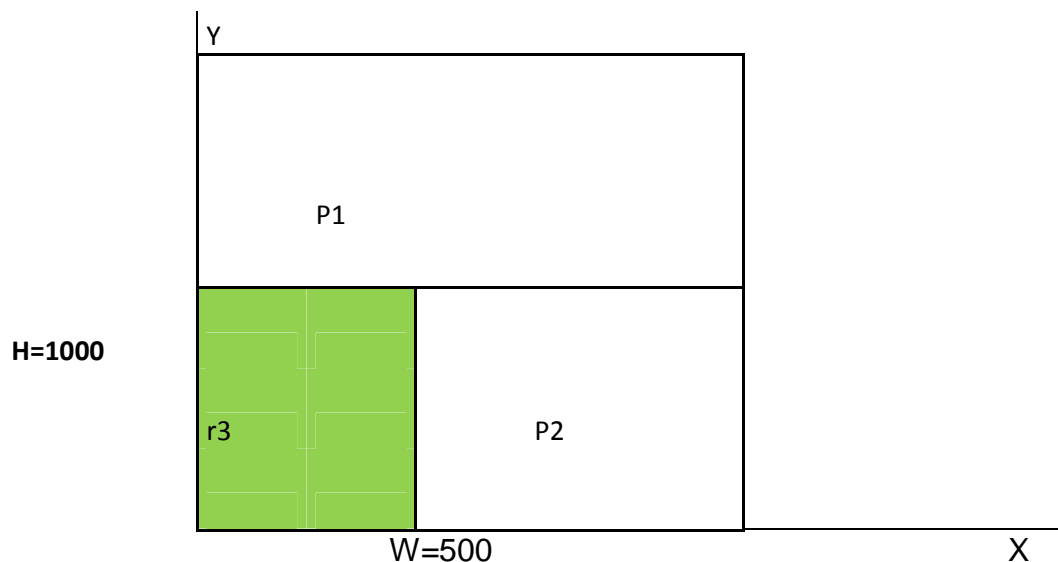
1- Calcul des positions faisables :

P	OX	OY	H	W	S
	0	0	1000	500	500000

Au début la feuille est vide on as une seule position de dimension H=1000 et W=500 dont l'origine est le coin inferieur gauche (BL)

Rectangle	Hi	Wi	Si	Fi	GN/ P
r2	400	100	40000	1	0
r3	500	200	100000	1	1
r4	700	200	140000	1	0
r6	500	200	10000	1	1
r7	400	200	80000	1	0
r8	300	200	60000	1	0
r9	800	100	80000	1	0
r10	700	200	140000	1	0

Le candidat est r3



Chapitre 5 Application pratique

L'ensemble $R=\{r2,r4,r6,r7,r8 ,r9,r10\}$

L'ensemble $I=\{r5,r1,r3\}$

Etape 4 :

1-calcul des positions faisables

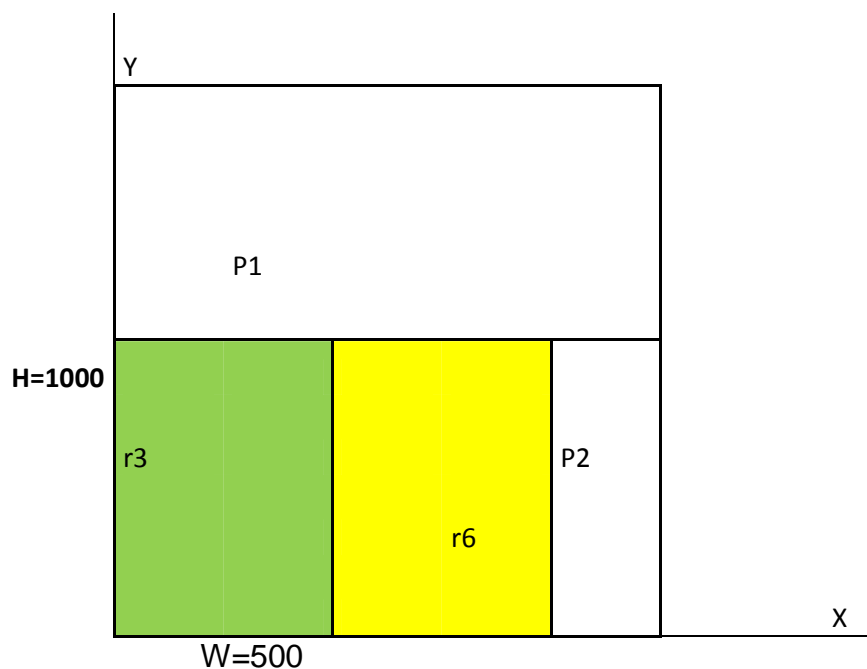
En plaçant r3 (500,200) deux positions faisables sont créées

P1	OX	OY	H	W	S
	0	500	500	500	250000
P2	OX	OY	H	W	S
	200	0	500	300	150000

Rectangle	hi	wi	Si	Fi/P1	GN/ P1	Fi/P2	GN/ P2
r2	400	100	40000	1	0	1	0
r4	700	200	140000	0	0	0	0
r6	500	200	100000	1	1	1	1
r7	400	200	80000	1	0	1	0
r8	300	200	60000	1	0	1	1
r9	800	100	80000	0	0	0	0
r10	700	200	140000	0	0	0	0

Le candidat r6 peut être placé dans P1 ou P2

On choisi le placement ou le reste de la surface est minimale c'est P2



Chapitre 5 Application pratique

L'ensemble $R=\{r2,r4,r7,r8,r9,r10\}$

L'ensemble $I=\{r5,r1,r3,r6\}$

Etape 5 :

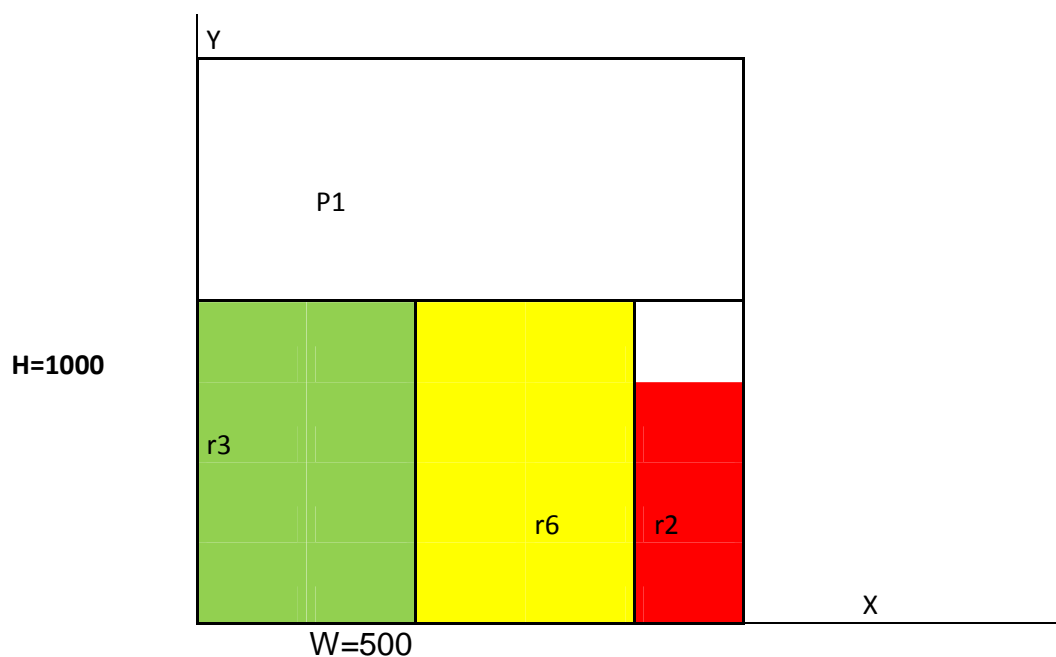
1- calculs des positions faisables :

En plaçant $r6(500,200)$ deux positions faisables sont créées

P1	OX	OY	H	W	S
	0	500	500	500	250000
P2	OX	OY	H	W	S
	400	0	500	100	50000

Rectangle	hi	wi	Si	Fi/P1	GN/ P1	Fi/P2	GN/ P2
r2	400	100	40000	1	0	1	2
r4	700	200	140000	0		0	
r7	400	200	80000	1	0	0	
r8	300	200	60000	1	0	0	
r9	800	100	80000	0		0	
r10	700	200	140000	0		0	

Le candidat à placer est r2 avec $G_n= 2$



L'ensemble $R=\{r4,r7,r8 ,r9,r10\}$

Chapitre 5 Application pratique

L'ensemble $I=\{r5,r1,r3,r6,r2\}$

Etape 6 :

1-calculs des positions faisables :

En plaçant $r2(500,200)$ deux positions faisables sont créés

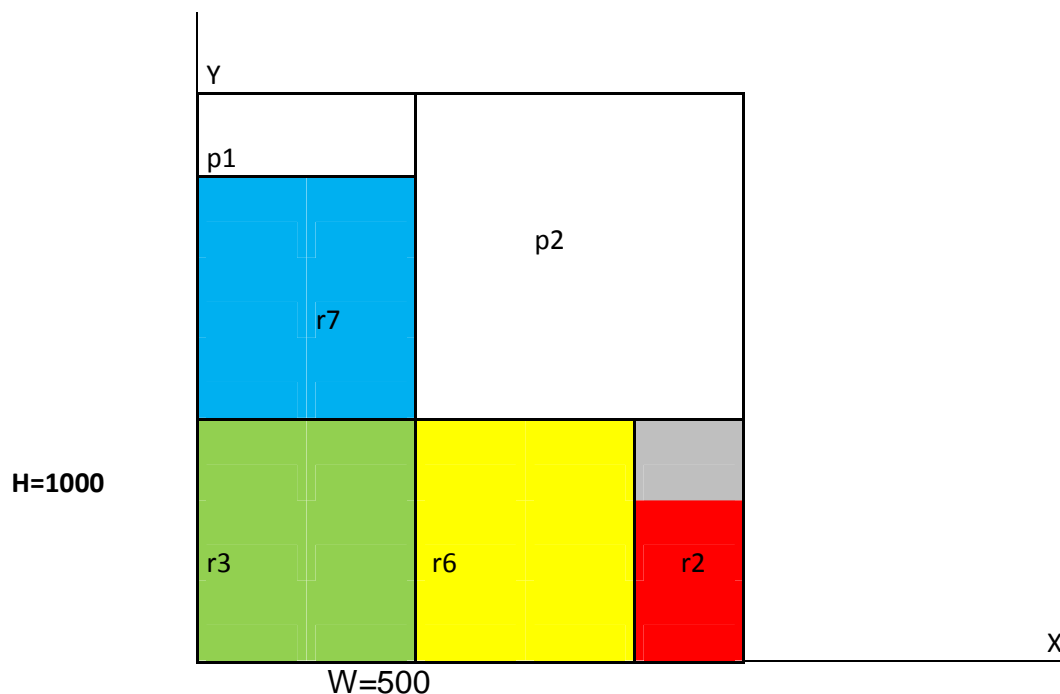
P1	OX	OY	H	W	S
	0	500	500	500	250000
P2	OX	OY	H	W	S
	400	400	100	100	10000

La position 2 est éliminée sa surface est inférieure a toutes les surfaces des rectangles presents. la position faisables est :

P1	OX	OY	H	W	S
	0	500	500	500	250000

Rectangle	Hi	Wi	Si	Fi/P1	GN/ P1
r4	700	200	140000	0	
r7	400	200	80000	1	0
r8	300	200	60000	1	0
r9	800	100	80000	0	
r10	700	200	140000	0	

r7 est candidat a placer



L'ensemble $R=\{r4,r8,r9,r10\}$

L'ensemble $I=\{r5,r1,r3,r6,r2,r7\}$

Etape 7 :

Chapitre 5

Application pratique

1-calculs des positions faisables

En plaçant r7(400,200) deux positions faisables sont créés.

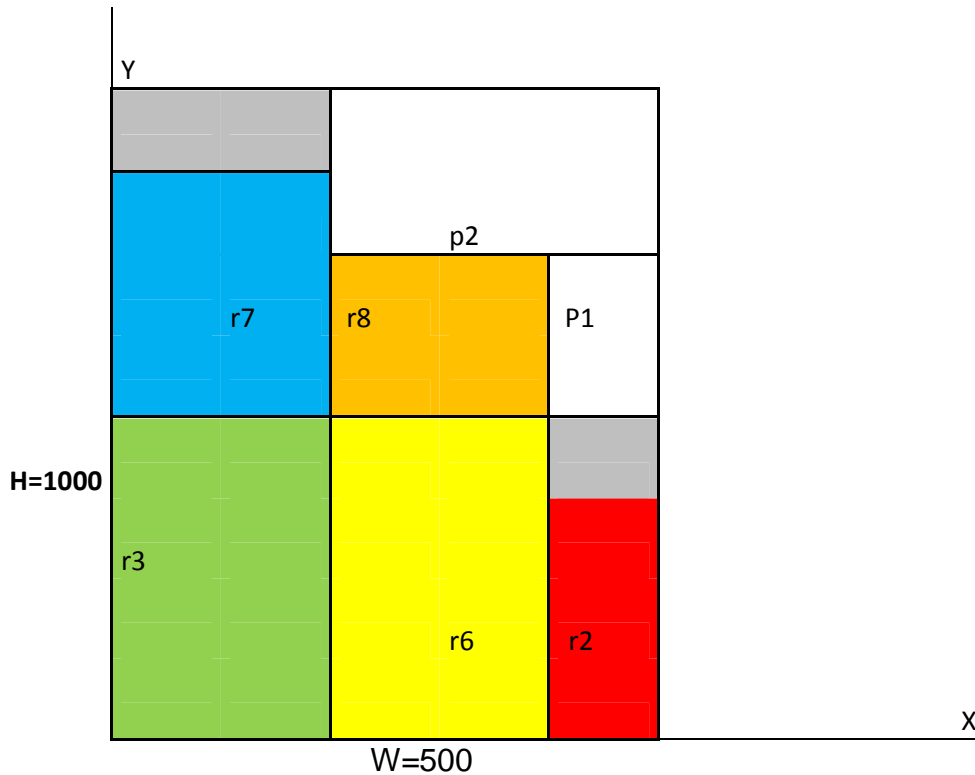
P1	OX	OY	H	W	S
	0	900	100	200	20000
P2	OX	OY	H	W	S
	200	500	500	300	150000

La position 1 est éliminée sa surface est inférieur a toutes les surfaces des rectangles présents.

La position faisable

P2	OX	OY	H	W	S
	200	500	500	300	150000

Rectangle	Hi	Wi	Si	Fi/P1	GN/ P1
r4	700	200	140000	0	
r8	300	200	60000	1	0
r9	800	100	80000	0	
r10	700	200	140000	0	



L'ensemble R = {r4,r9,r10}

L'ensemble I={r5,r1,r3,r6,r2,r7,r8}

Chapitre 5 Application pratique

Etape 8 :

1-calculs des positions faisables :

En plaçant r8(300,200) deux positions faisables sont créées

P1	OX	OY	H	W	S
	400	500	300	100	30000
P2	OX	OY	H	W	S
	200	800	300	200	60000

Rectangle	Hi	Wi	Si	Fi/P1	GN/ P1	Fi/P1	GN/ P1
r4	700	200	140000	0		0	
r9	800	100	80000	0		0	
r10	700	200	140000	0		0	

Aucune position n est faisable on ouvre une troisième feuille.

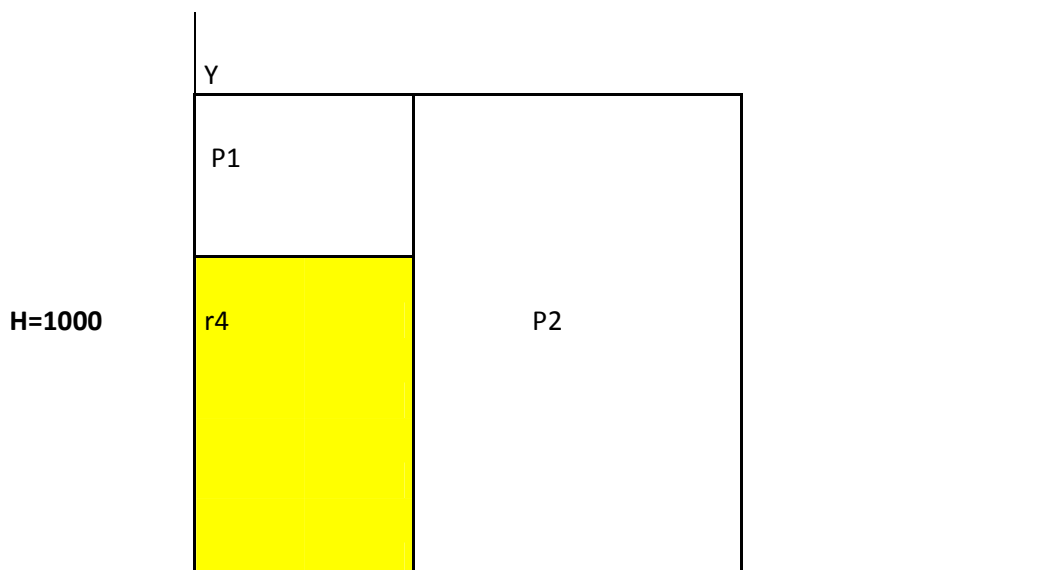
Etape 9 :

1-calculs des positions faisables :

P	OX	OY	H	W	S
	0	0	1000	500	500000

Au début la feuille est vide on a une seule position de dimension H=1000 et W=500 dont l'origine est le coin inférieur gauche(BL).

Rectangle	hi	wi	Si	Fi/P1	GN/ P1
r4	700	200	140000	1	0
r9	800	100	80000	1	0
r10	700	200	140000	1	0



Chapitre 5 Application pratique

W=500

X

Candidat à placer r4(700,200)

L'ensemble R={r9,r10}

L'ensemble I={r5,r1,r3,r6,r2,r7,r8,r4}

Etape 10 :

1-Calculs des positions faisables :

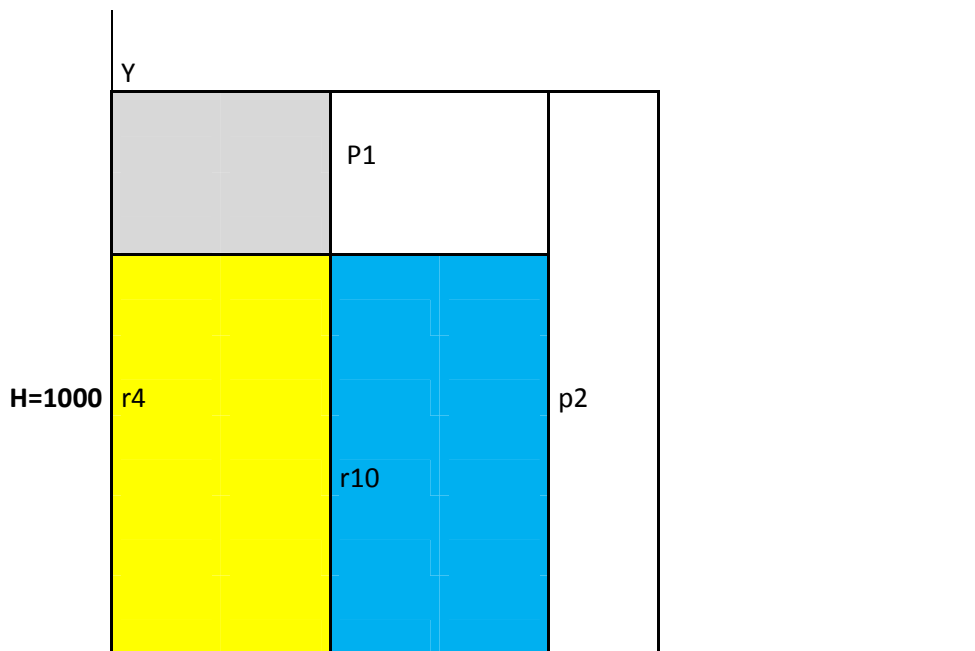
En plaçant r4=(700,200) deux position faisables sont créés

P1	OX	OY	H	W	S
	0	700	300	200	60000
P2	OX	OY	H	W	S
	200	0	1000	300	300000

P1 est éliminée parce qu'aucun rectangle ne peut être placé dans cette position

Rectangle	Hi	wi	Si	Fi/P2	GN/ P2
r9	800	100	80000	1	0
r10	700	200	140000	1	0

le candidat à placer est r10 qui a la plus grande surface



W=500

X

L'ensemble R={r9}

L'ensemble I={r5,r1,r3,r6,r2,r7,r8,r4 ;r10}

Chapitre 5 Application pratique

Etape 11

1-calculs des positions faisables :

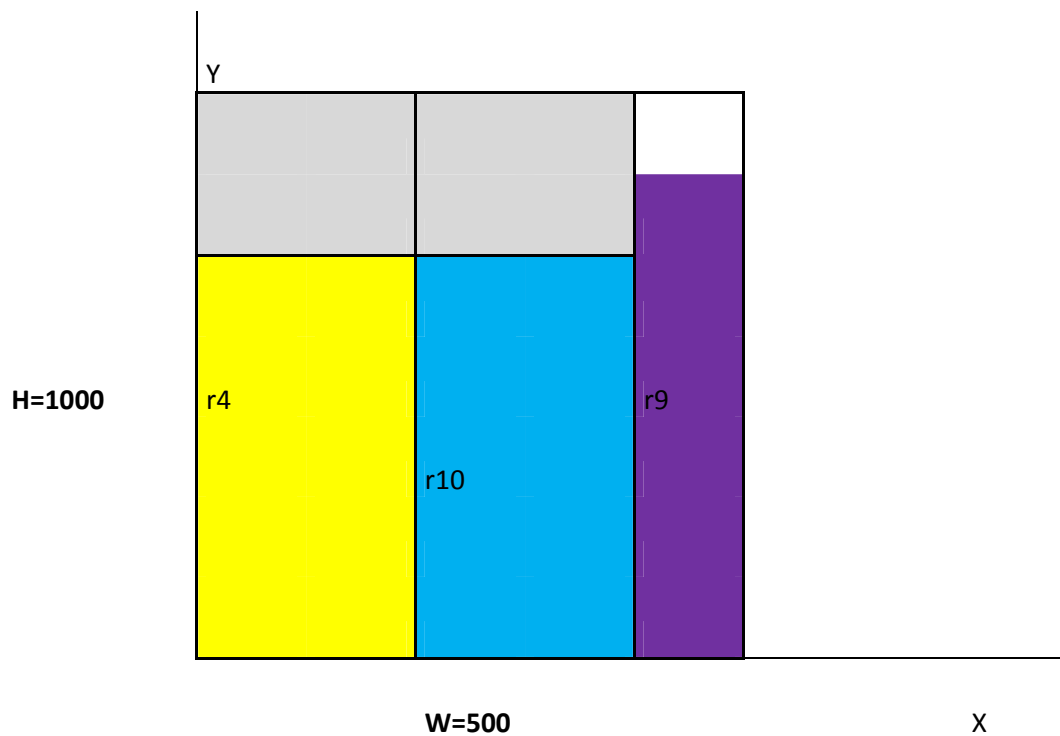
En plaçant r10(700,200) deux positions faisables sot créés

P1	OX	OY	H	W	S
	200	700	300	200	60000
P2	OX	OY	H	W	S
	400	0	1000	100	100000

P1 est éliminée parce qu'aucun rectangle ne peut être placée dans cette position

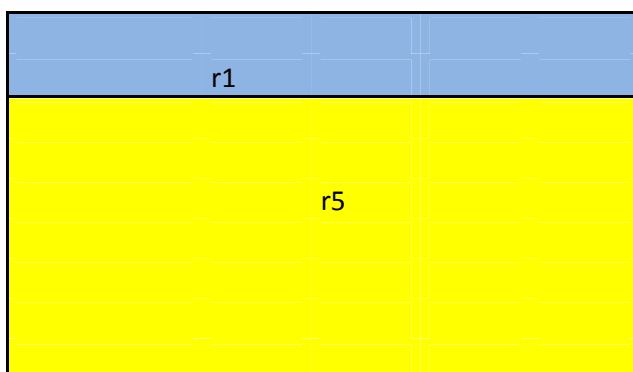
Rectangle	Hi	Wi	Si	Fi/P1	GN/ P1
r9	800	100	80000	1	0

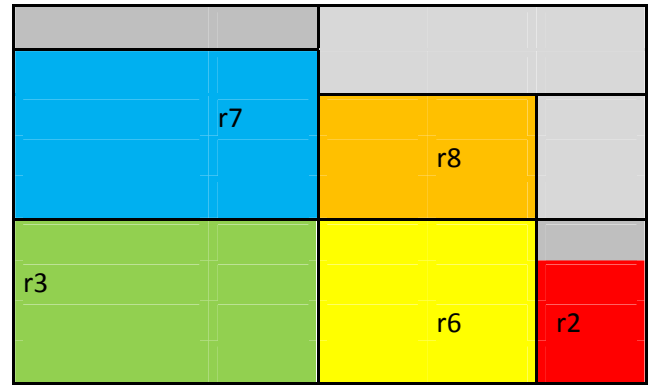
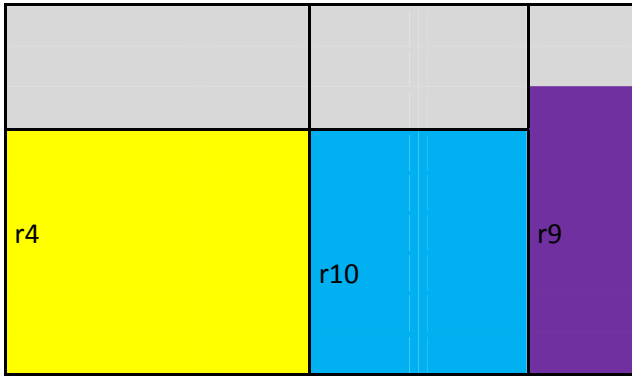
Le dernier rectangle est placé à la position P2



Conclusion

❖ La solution donnée par l'heuristique utilise trois feuilles :





- ❖ La solution optimale est égale à la surface totale des rectangles 12400 sur la surface d'une feuille 5000 ; on trouve le nombre de feuille optimal à utiliser égale à $\frac{12400}{5000} = 2.48$
- ❖ L'optimalité de la solution ρ est égale à la solution optimale sur la solution de l'heuristique $\frac{2.48}{3} = 0.83$ qui est proche de 1

4. Résolution de OptiCoup V

Afin de comparer les résultats obtenus avec l'heuristique LWF nous allons résoudre la même application en utilisant le Logiciel OptiCoup V

Dans le sous-menu "Unités" du menu "Outils", vérifier que l'unité est le millimètre.

2 - Saisie du matériau

Créer le matériau **MDF280**, correspondant à des panneaux de MDF , et auquel seront rattachés le panneau de 1000x500.

Pour cela, ouvrir le sous-menu "Matériaux des Panneaux" du menu "Stock" créer le matériau MDF280, en renseignant les différentes rubriques.

3 - Saisie du panneau

Ouvrir le sous-menu "Edition des Panneaux" du menu "Stock" et créer un panneau de 1000 x 500 dans le matériau MDF280.

4 - Saisie du Débit

Ouvrir le sous-menu "Nouveau Débit de Panneaux" du menu "Fichier"), et saisir les 7 lignes du débit.

Lors de la saisie de la première ligne, renseigner le matériau MDF280, et vérifier que la case "Sens du fil" est décochée. Ces paramètres seront reconduits automatiquement pour les lignes suivantes.

5 - Choix des Paramètres

Ouvrir le sous-menu "Paramètres d'Optimisation" du Menu "Optimisation", et fixer les paramètres suivants :

- @ Epaisseur de la lame (Panneau) : 4 mm.
- @ Affranchissements généraux (*Hauteur & Largeur*) : 0 mm.
- @ Mode d'optimisation : Rapide ou Avancé 2.
- @ Sens de la première coupe : Indifférent.

6 - Calcul d'optimisation

Pour lancer le calcul, cliquez sur le sous-menu "Optimiser" du menu "Optimisation", ou sur l'icône symbolisée par deux roues crantées.

En cours de calcul, un message vous indique la progression. La durée du calcul varie en fonction de la puissance du micro-ordinateur, mais dans ce cas simple, elle ne devrait pas excéder quelques secondes en mode Avancé 2.

Au terme du calcul en mode Avancé 2, les 10 Pièces sont placées dans 4 panneaux comme suit :

Résultat en mode Avancé 2

Numéro	Matériau	Dimension	Référence	Nombre	Pièces	Linéaire de découpe	Taux de chutes	Taux de chutes net
1 / 4	CTP, 5,00	1000.00 × 500.00		1	2	1.10 m	12.00 %	12.00 %
2 / 4	CTP, 5,00	1000.00 × 500.00		1	3	2.60 m	28.00 %	28.00 %
3 / 4	CTP, 5,00	1000.00 × 500.00		1	4	2.09 m	28.00 %	5.03 %
4 / 4	CTP, 5,00	1000.00 × 500.00		1	1	1.10 m	84.00 %	4.80 %

Plan de coupe 1 / 4
CTP, 5.00, 1000.00 × 500.00

Démon

Démon

Démon

Chapitre 5

Application pratique

The screenshot shows a software interface for panel optimization. The main window contains a table of cut data, a tree view of the layout, and a detailed view of a specific cut plan.

Numéro	Matériau	Dimension	Référence	Nombre	Pièces	Linéaire de découpe	Taux de chutes	Taux de chutes net
1 / 4	CTP, 5,00	1000.00 × 500.00		1	2	1.10 m	12.00 %	12.00 %
2 / 4	CTP, 5,00	1000.00 × 500.00		1	3	2.60 m	28.00 %	28.00 %
3 / 4	CTP, 5,00	1000.00 × 500.00		1	4	2.09 m	28.00 %	5.03 %
4 / 4	CTP, 5,00	1000.00 × 500.00		1	1	1.10 m	84.00 %	4.80 %

Nombre de panneaux utilisés
Nombre de plans de coupe
Surface totale des panneaux: 2.00
Taux de chutes: 38.0
Taux des chutes non réutilisables: 12.4

Plan de coupe 2 / 4
CTP, 5,00, 1000,00 × 500,00

Démo

Fichier Affichage Optimisation Stock Outils Langue Fenêtre ?

Numéro	Matériau	Dimension	Référence	Nombre	Pièces	Linéaire de découpe	Taux de chutes	Taux de chutes net
1 / 4	CTP, 5,00	1000.00 × 500.00		1	2	1.10 m	12.00 %	12.00 %
2 / 4	CTP, 5,00	1000.00 × 500.00		1	3	2.60 m	28.00 %	28.00 %
3 / 4	CTP, 5,00	1000.00 × 500.00		1	4	2.09 m	28.00 %	5.03 %
4 / 4	CTP, 5,00	1000.00 × 500.00		1	1	1.10 m	84.00 %	4.80 %

Nombre de panneaux utilisés
 Nombre de plans de coupe
 Surface totale des panneaux
 Taux de chutes
 Taux des chutes non réutilisables

Superposition
 202.00
 406.00
 610.00
 Changement de sens de coupe
 Démo
 Démo
 Chute non réutilisable : Démo
 Chute : Démo

Plan de coupe 3 / 4
 CTP, 5,00, 1000,00 × 500,00

The screenshot shows a software interface with a menu bar (Fichier, Affichage, Optimisation, Stock, Outils, Langue, Fenêtre, ?) and a toolbar. On the left, there is a sidebar with statistics: 'Nombre de panneaux utilisés', 'Nombre de plans de coupe', 'Surface totale des panneaux' (2.00), 'Taux de chutes' (38.0), and 'Taux des chutes non réutilisables' (12.4). Below this is a legend for 'Chute non réutilisable : Démo' and 'Chute : Démo'. The main area contains a table with the following data:

Numéro	Matériau	Dimension	Référence	Nombre	Pièces	Linéaire de découpe	Taux de chutes	Taux de chutes net
1 / 4	CTP, 5,00	1000.00 × 500.00		1	2	1.10 m	12.00 %	12.00 %
2 / 4	CTP, 5,00	1000.00 × 500.00		1	3	2.60 m	28.00 %	28.00 %
3 / 4	CTP, 5,00	1000.00 × 500.00		1	4	2.09 m	28.00 %	5.03 %
4 / 4	CTP, 5,00	1000.00 × 500.00		1	1	1.10 m	84.00 %	4.80 %

Below the table, a diagram titled 'Plan de coupe 4 / 4' shows a large rectangle representing a sheet of material. A smaller rectangle is cut out from the bottom-left corner, labeled 'Démo'. The remaining area is also labeled 'Démo'.

❖ La solution optimale est égale à la surface totale des rectangles 12400 sur la surface d'une feuille 5000 ; on trouve le nombre de feuille optimal à utiliser égale à

$$\frac{12400}{5000} = 2.48$$

❖ L'optimalité de la solution ρ est égale à la solution optimale sur la solution de l'heuristique $\frac{2.48}{4} = 0.62$ qui est proche de 1.

Conclusion

Au vu des résultats obtenus par les deux méthodes, on remarque que l'heuristique LWF donne de meilleurs résultats que OptiCoupV, en effet

Le nombre de feuille ainsi que le taux de chute est meilleur avec LWF qui utilise trois feuilles alors que OptiCoup V utilise 4. le calcul du paramètre d'optimalité ρ confirme cette conclusion puisque $\rho_{LWF} = 0.83 > \rho_{opticoupV} = 0.62$.

Conclusion générale

Conclusion Générale

Les problèmes de placement sont des problèmes d'optimisation combinatoire avec de nombreuses applications dans divers domaines tels que l'industrie de l'acier, du vêtement et la conception des circuits intégrés...

Le domaine de placement attire de nombreux chercheurs et praticiens. En fonction des applications, dans le domaine théorique beaucoup d'articles sont dédiés à ce domaine ou des heuristiques sont développées pour résoudre ce problème et améliorer l'obtention de solution aussi proche de l'optimal dans un temps acceptable. Côté pratique le marché est inondé de logiciels paramétrés permettent aussi de résoudre pas mal de ces problèmes avec des solutions assez satisfaisantes.

Dans notre mémoire, après avoir abordé l'aspect théorique des problèmes de placement, on s'est attardé au placement de forme rectangulaire orthogonal, ou une liste de rectangles doit être placée dans une surface désignée, de sorte qu'aucuns de ces rectangles ne se chevauchent et tous les rectangles sont placés en parallèle à l'axe de la surface. L'objectif est surtout de maximiser le taux d'utilisation de la surface de travail pour moins de chutes.

Alors que plusieurs algorithmes et heuristiques se présentent, après plusieurs recherche et compréhension, notre étude nous a conduis à choisir une heuristique LWF, que nous avons développé et appliqué à cas pratique, puis on a chois un logiciel OptiCoup V pour se familiariser avec l'outil informatique pour la résolution de ce problème, ca nous a aussi permet de comparer les résultats obtenus par les deux méthodes de résolution.

Ce travail nous a permet de traiter un cas pratique, découvrant ainsi la méthodologie de résolution des problèmes, le passage par la modélisation de l'aspect pratique à la résolution mathématiques pour finir sur une solution qui n'est pas toujours optimal tout en essayant de s'en approché, vu que les problèmes abordés sont pratiquement difficile au sens de la complexité. La partie informatique reste importante pour finaliser la solution. Notre travail reste théorique, même si l'application traitée a été réalisé manuellement, aussi avec un logiciel pour voir le comportement. Vu la complexité de l'heuristique LWF que nous avons développé et qui donne des résultats assez intéressant, La programmation de cette méthode est vivement souhaité et qui peut faire l'objet de sujet de mémoire intéressant dans l'avenir.

Bibliographie

- [1] Abdesslem LAYEB, Utilisation des Approches d'Optimisation Combinatoire pou La Vérification des Applications Temps Réel. Thèse de Doctorat, Université Mentouri de Constantine 2010
- [2] B. R. Tebbal mohamed. Etude de problème de rangement ouvert avec conflits. Master's thesis, 2016.
- [3] Mostepha, R : Résolution de problèmes d'optimisation combinatoire par systèmes artificiels auto-organisés. Thèse de magister, Université Mentouri de Constantine ,2008.
- [4] J. Desrosiers, Y. Dumas, and F. Soumis. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences*, 6(3-4) :301–325, 1986.
- [5] J. P. Heuristics. Intelligent search strategies for computer problem solving addison, 1984.
- [6]] C. Guéret, C. Prins, and M. Sevaux. Programmation linéaire. eyrolles. Technical report, ISBN 2-212-09202-4, 365 pages (in French), 2000.
- [7] R. C. Corrêa, A. Ferreira, and S. C. Porto. Selected algorithmic techniques for parallel optimization. In *Handbook of Combinatorial Optimization*, pages 1879–1928. Springer, 1998.
- [8] P. Schwerin and G. Wäscher. The bin-packing problem : A problem generator and some numerical experiments with ffd packing and mtp. *International Transactions in Operational Research*, 4(5-6) :377–389, 1997.
- [9]] P. Siarry. La méthode du recuit simulé : théorie et applications. *Automatique-productique informatique industrielle*, 29(4-5) :535–561, 1995.
- [10] A. Hanset, H. Fei, O. Roux, D. Duvivier, and N. Meskens. Ordonnancement des interven- tions chirurgicales par une recherche tabou : Exécutions courtes vs longues. *Logistique et Transport LT* 2007, 2007.
- [11] J. H. Holland. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2) :88–105, 1973.
- [12] J. L. Bruno and P. J. Downey. Probabilistic bounds for dual bin-packing. *Acta Informatica*, 22(3) :333–345, 1985.
- [13] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4) :28–39, 2006.
- [14] D. Datta and J. R. Figueira. A real-integer-discrete-coded particle swarm optimization for design problems. *Applied Soft Computing*, 11(4) :3625–3633, 2011.
- [15]] D. Liu, K. C. Tan, S. Huang, C. K. Goh, and W. K. Ho. On solving multiobjective bin packing problems using evolutionary particle swarm optimization. *European Journal of Operational Research*, 190(2) :357–382, 2008.
- [16] T. Sofiane. Résolution de problemes de bin packing a une dimension par la programmation dc. Master's thesis, Université abederrahmane Mira Bejaia, 2013/2014.
- [17] H. Dyckhoff et U. Finke, *Cutting and Packing in Production and Distribution; A Typology and Bibliography*, ed. Physica-verlag, A Springer-Verlag, 1992.
- [18] G. Wäscher, H. Haussner, H. Schumann, An improved typology of cutting and packing problems. *European Journal of Operational Research* 83 {3), 1109-1130, 2007.
- [19] A. Lodi, S. Martello, D. Vigo, Heuristic and Metaheuristic Approaches for a Class of Twodimensional Bin packing problems. *INFORMS Journal on Computing* 11, 345-357, 1999.
- [20] M. Hifi, I. Kacem, S. Negre, L. Wu, A Linear Programming Approach for the Three-Dimensional Bin-Packing Problem. *Electronic Notes in Discrete Mathematics*, 36, 993- 1000, 2010.
- [21] P. Gilmore, R. Gomory, A linear programming approach to the cutting stock problem. *Operational Research*, 9{6), 849 - 859, 1961.
- [22] Azzouz Hamza.-Mlata Ibrahim Elkhalil.Yahyaoui. Zakaria, «Modélisation et optimisation de

l'espace en utilisant le « Nesting »,» Projet Fin d'Etude Master 2 Genie Industriel, Université Aboubekr Belkaid de Tlemcen, 2016/2017.

[23] C. K. B. YTHIP, «Algorithmes Heuristiques et Evolutionnistes : Application à la Résolution du Problème de Placement de Formes Irrégulières,» Pour obtenir le grade de Docteur, L'UNIVERSITE DES SCIENCES ET TECHNOLOGIQUES DE LILLE, 9 octobre 1998.

- [24] Dagli CH, Poshyanonda P. New approaches to nesting rectangular patterns. Journal of Intelligent Manufacturing 1997;3(3):177--90.