

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.

Cette discipline reçoit son nom pour la première fois en Angleterre et prouve son efficacité par un rapprochement de scientifiques et de militaires chargés de préparer les grandes décisions liées aux opérations militaires pendant la seconde guerre mondiale.

L'application de la Recherche Opérationnelle s'est élargie dans ces dernières décennies 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.

Le problème de chargement des conteneurs de placement sont des variantes d'un problème d'intérêt majeur qui est le problème de Bin Packing. 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 ranger des articles dans des conteneurs de la manière la plus économique possible. Les contraintes liées aux objets et aux conteneurs 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 chargement des conteneurs qui concerne des boîtes de formes rectangulaire par une heuristique récente bien défini qui s'adapte à ce type de problème, puis le résoudre aussi à l'aide d'un logiciel disponible sur le marché.

Organisation du mémoire

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

Chapitre I : l'introduction à 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 II : le 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 III : généralité sur le problème de placement

Dans ce chapitre pour continuer notre concept, nous décrirons les différents types de problèmes de placement et les contraintes liées au problème.

Chapitre IV : problème de chargement de conteneur

On définit dans ce chapitre, le problème de chargement de conteneur en citant les différentes contraintes liées au problème.

Chapitre V : Application pratique

Nous avons abordé les solutions du problème : l'heuristique de gourmande et la méthode des matrices que nous avons appliqués à un problème pratique.

Une autre application a été résolue en utilisant le logiciel « container load calculator ».

Puis nous avons étudié un article traitant de l'état de l'art de ce problème

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

Chapitre I :
Introduction
à l'optimisation de combinatoire

Introduction

L'optimisation combinatoire appelée aussi optimisation discrète, est une branche de l'optimisation 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'ingénierie 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]

Les stratégies d'optimisation combinatoire font aujourd'hui partie des moyens auxquels les dirigeants d'entreprises ont recours pour faciliter la prise de décision ou la conception de solutions devant un cas de figure inhabituel ou complexe. Elles ont maintes fois prouvé leur efficacité que ce soit dans le secteur de la production, de l'organisation ou encore en cas de soucis économiques ou informatiques.

I.1. Formulation d'un problème d'optimisation combinatoire

Un problème d'optimisation combinatoire se définit à partir d'un triplet (E, p, f) tel que:

- E est un ensemble discret appelé espace de solution ou encore espace de recherche.
- p est un prédicat sur E, i.e. une fonction de E dans {Vrai, Faux}.
- $f : E \rightarrow \mathbb{R}$ associe à tout élément $x \in E$ un coût $f(x)$, f est appelé fonction objectif ou fonction de coût.

$$\max_{x \in E} f(x) = - \min_{x \in E} (-f(x))$$

L'ensemble des solutions admissibles du problème, noté E_a , est défini tel que $E_a = \{x \in E \text{ tel que } p(x) \text{ est Vrai}\}$.

I.2 Exemples de problème d'optimisation combinatoire

- **Problème de remplissage optimal (conteneurs et sac-a-dos et placement).**
- **Problème de découpe.**
- **Problème de tournées de véhicules.**
- **Problème de transport.**
- **Problème de conception de l'emploi du temps.**
- **Problème d'ordonnancement.**
- **Problème d'affectation...etc.**

I.3 Généralités sur la Théorie de la Complexité [3]

La Théorie de la complexité constitue l'étude de la difficulté des problèmes (traités/résolus par des algorithmes). Lorsqu'un algorithme **A** résout un problème, on peut toujours se demander s'il y a un algorithme **A!** Qui résout aussi le problème mais qui est plus efficace. Il s'agit donc, d'abord d'évaluer, comparer, optimiser, les performances des algorithmes.

Dans le but de mieux comprendre comment les problèmes se placent les uns par rapport aux autres, la théorie de la complexité établit des hiérarchies de difficulté entre les problèmes algorithmiques, dont les niveaux sont appelés des « *classes de complexité* ». Ces hiérarchies comportent des ramifications, suivant que l'on considère des calculs déterministes – l'état suivant du calcul est « *déterminé* » par l'état courant – ou non déterministes.

Exemples

- Comparaison des algorithmes de tri : quel algorithme de tri vaudrait-il mieux utiliser pour trier les éléments d'un tableau ?
- Si on lance le calcul de la factorielle de 100, combien de temps faudra-t-il pour attendre le résultat?

- **Définition 01 [4]**

La Complexité d'un algorithme **A** est une fonction $CA(N)$ donnant le nombre d'instructions exécutées par **A** dans le pire des cas pour une donnée de taille **N** (le paramètre de complexité). La complexité est basée généralement sur le pire cas, afin de borner le temps d'exécution de l'algorithme.

I.3.1 Types de complexité

Deux types de complexité peuvent être cités:

- **La complexité méthodologique** : qui exprime une fonction du nombre d'opérations élémentaires de calcul effectuées par la méthode ou par l'algorithme de résolution en fonction du nombre des données du problème traité.

- **La complexité problématique** : liée à la difficulté du problème à résoudre et au nombre des opérations élémentaires qu'un algorithme déterministe peut effectuer pour trouver l'optimum en fonction de la taille du problème.

I.3.2 Les différentes classes de complexité [6]

On distingue deux principales classes des complexités : La classe des problèmes **P** et la classe des problèmes **NP**, nous développerons brièvement ces notions.

- Classe des problèmes P

Un Problème appartient à la classe **P**, si et seulement s'il existe un algorithme déterministe en temps polynomial (par rapport à la taille de l'instance n), permettant de répondre OUI au problème (problème de décision). On qualifie ce problème de polynomial, et on dit qu'il est facile.

Exemples

- *Le test de primalité AKS* est un algorithme qui montre que le problème de savoir si un entier est premier ou non peut être résolu par un algorithme polynomial.
- Programmation linéaire : Les algorithmes de points intérieurs permettent de classer la programmation linéaire dans la classe **P**.

- Classe des problèmes NP

La classe **NP** réunit des problèmes pour lesquels la vérification d'une solution peut être décidée par un algorithme non-déterministe en un temps polynomial par rapport à la taille de l'instance n .

Intuitivement, les problèmes **NP** sont tous les problèmes qui peuvent être résolus en énumérant l'ensemble des solutions possibles et en les testant avec un algorithme polynomial.

Il existe deux catégories de cette classe, les problèmes **NP-complet** et les problèmes **NP-difficiles**.

Exemple

- Le problème du voyageur de commerce appartient à la classe des problèmes NP.
- **Les problèmes NP-complet** : cette classe contient la partie la plus difficile de NP. Un problème d'existence est dit NP-complet s'il appartient à NP et si tout problème de NP peut être transformé de façon polynomiale en ce problème. Ce sont les problèmes les plus étudiés. Ceci parce que beaucoup de problèmes intéressants sont NP-complet et que l'on ne sait pas résoudre un problème NP-complet efficacement à cause du non déterminisme.

Un problème de décision est NP-complet s'il vérifie les propriétés suivantes :

- Il est possible de vérifier une solution efficacement (en temps polynomial) ;
- Tous les problèmes de la classe NP se ramènent à celui-ci via une réduction polynomiale ; cela signifie que le problème est au moins aussi difficile que tous les autres problèmes de la classe NP.

Bien qu'on puisse vérifier rapidement toute solution proposée d'un problème NP-complet, on ne sait pas en trouver efficacement.

Exemples

- *Problème du voyageur de commerce,*
- *Problème du sac à dos,*
- *Problème de coloration de graphe ...*

La notion de réduction polynomiale est très importante pour établir le NP-complet d'un problème de décision donné.

- **Les problèmes NP-difficiles** : Un problème d'optimisation combinatoire est dit NP-difficile si et seulement si le problème d'existence associé est NP-complet.

➤ **Remarque**

Les problèmes NP-complet appartiennent aux problèmes NP-difficile par contre un problème NP-difficile n'est pas nécessairement dans NP.

I.4 Les Méthodes de résolution d'un problème d'optimisation combinatoire [3]

Les méthodes de résolution des problèmes d'optimisation combinatoire sont classées en deux catégories: les méthodes exactes et les méthodes approchées.

I.4.1 Les Méthodes de résolution exactes

Les méthodes de résolution exactes ou (complètes) examinent de manière implicite la totalité de l'espace de recherche. Ces méthodes de résolution exactes sont nombreuses, elles ont l'avantage de produire une ou plusieurs solutions dont l'optimalité est garantie lorsqu'aucune contrainte de temps n'est donnée.

Néanmoins, le temps de calcul nécessaire pour atteindre une solution optimale peut devenir vite prohibitif; ce malgré les diverses techniques et heuristiques qui ont été développées pour accélérer l'énumération des solutions.

Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème. On trouve parmi ces méthodes : les méthodes de la programmation linéaire, la méthode de Branch and Bound, la méthode hongroise, la méthode de la programmation dynamique,...etc.

Nous allons présenter ici deux méthodes exactes :

➤ **Programmation Linéaire**

Les problèmes de programmation linéaires (PL) sont des problèmes d'optimisation ou la fonction objectif, et les contraintes sont toutes linéaires. La programmation linéaire désigne également la manière de résoudre les problèmes linéaires. Cette dernière est un domaine central de l'optimisation, car les problèmes de PL sont les problèmes

d'optimisation les plus facile toutes les contrainte y étant linéaire beaucoup de problèmes réelle de recherche opérationnelle peuvent être exprimé comme problèmes de PL.

- **La technique de la programmation linéaire**

La mise en œuvre de la technique de programmation linéaire peut être subdivisée en cinq étapes.

Première étape

Identification du problème comme étant solvable par la programmation linéaire. Cette identification est le fruit d'une expérience dans la modélisation mathématique de problème.

Deuxième étape

Formulation du problème réel avec utilisation d'un modèle mathématique linéaire. Elle se fait en collaboration avec le décideur qui pose le problème.

Troisième étape

Résolution du problème théorique en utilisant des techniques algorithmiques.

Quatrième étape

Détermination d'une solution réelle à partir de la solution théorique.

Cinquième étape

Vérification de la validité de la solution et modification, si nécessaire de la formulation mathématique. Cette étape permet d'affiner le modèle afin d'apporter une solution acceptables par le décideur.

- Méthode graphique

Cette méthode consiste à représenter l'ensemble des contraintes sur un repère cartésien. Les contraintes ou apparaissent des inégalités et des égalités correspondent géométriquement à des demi-plans. Intersection de ces demi-plans donnera ensemble des variables satisfaisant à toutes les contraintes. L'ensemble des contraintes est un polygone convexe.

- La méthode du simplexe

La méthode du simplexe est un algorithme de recherche d'une solution optimale d'un programme linéaire donné. La mise en œuvre de la méthode du simplexe peut être divisée en trois étapes :

Première étape : Mettre le modèle sous forme standard en y introduisant des variables d'écart qui ont pour rôle de transformer les inégalités en égalités.

Deuxième étape : Etablir le premier tableau de simplexe (tableau à l'origine).

Troisième étape : Procéder une série d'itérations sur les tableaux de simplexe aboutissant soit : la solution optimale ou une infinité de solution ou le modèle n'admet pas de solution.

Algorithme du simplexe

Pas 0 : *initialisation*

Mettre le problème sous forme standard (contraintes d'égalités et variables positives).

Trouver une première solution de base réalisable (utiliser la méthode section suivante si nécessaire).

Si aucune solution de base réalisable n'existe, alors **STOP**. Le problème est irrésoluble

Pas 1: *Choix de la variable entrante*

- ✓ Choisir comme variable entrante une variable hors base qui améliore la valeur courante de l'objectif.
- ✓ Si aucune variable hors base n'est candidate, alors **STOP**. La solution de base courante est une solution optimale.

- Méthode dual

L'optimisation d'un programme linéaire à l'aide de la méthode de simplexe nous oblige parfois à introduire des variables artificielles pour obtenir une solution de départ lorsque les contraintes sont de type $AX \geq b$ ($b \geq 0$).

La méthode des deux phases et la M-méthode ont été utilisées pour optimiser un programme linéaire ayant des variables artificielle, afin de réduire la taille de PL et le nombre d'itération en générale, on utilise l'algorithme dual du simplexe (qui n'utilise pas de variables artificielles).

- Algorithme dual de simplexe

Pas 0 : *initialisation*

- Mettre le problème sous forme standard (contraintes d'égalités et variables positives)
- Trouver une solution de base dual-réalisable (tous les coûts relatifs c_j sont positifs (négatifs) dans le cas d'un problème de minimisation (maximisation))

Pas 1 : *Choix de la variable sortante*

- Choisir comme variable sortante une variable en base dont la valeur est strictement négative. Soit x_{ir} cette variable, br la valeur de cette variable et r le numéro de la contrainte qui donne cette valeur.
- Si aucune variable de base n'est strictement négative, alors STOP. La solution de base dual réalisable courante est primal réalisable et optimale.

➤ La Programmation Dynamique

La Programmation Dynamique est une méthode exacte de résolution de problèmes d'optimisation, due essentiellement à R. Bellman (1957). Bien que très puissante, son cadre d'application est relativement restreint, dans la mesure où les problèmes qu'elle traite doivent vérifier un principe dit *principe d'optimalité* qui stipule qu'une solution optimale d'un problème de taille n peut s'exprimer en fonction de la solution optimale de problèmes de taille inférieure à n .

La programmation dynamique consiste à placer le problème dans une famille de problèmes de même nature mais de difficulté différente, puis à trouver une relation de récurrence liant les solutions optimales de ces problèmes. Elle est basée sur le principe de Richard BELLMAN « *Toute sous politique d'une politique optimale est optimale* ».

I.4.2 Les méthodes approchées [7]

Typiquement ce type de méthodes, dites **heuristiques** est particulièrement utile pour les problèmes nécessitant une solution en temps réel (ou très court) ou pour résoudre des problèmes difficiles sur des instances numériques de grande taille. Elles peuvent aussi être utilisées afin d'initialiser une méthode exacte.

Parmi ces méthodes, on distingue les heuristiques ciblées sur un problème particulier et les **méta-heuristiques** plus puissantes et adaptables pour résoudre un grand nombre de problèmes.

Les méthodes approchées permettent d'obtenir une solution de bonne qualité (c'est-à-dire assez proche de l'optimum) dans un contexte de ressources (temps de calcul et/ou mémoire) limitées. Dans ce cas l'optimalité de la solution ne sera pas garantie.

Les heuristiques et les méta-heuristiques sont le compromis entre le temps de résolution et les qualités des résultats.

Quelques heuristiques : les heuristiques gloutonnes, la méthode de la descente,...etc.

Quelques méta-heuristiques : méthode de recherche Tabou, méthode de recuit simulé, les algorithmes de colonie de fourmis, les algorithmes génétiques...etc.

Nous allons présenter quelques méta-heuristiques et heuristiques :

➤ **La Recherche Tabou**

Bien que son origine remonte à 1977, la Recherche Tabou n'est proposée qu'au milieu des années 80 par Fred Glover. Cette méthode, développée pour résoudre des problèmes combinatoires, la plupart NP-difficiles, propose de surmonter le problème des optima locaux par l'utilisation d'une mémoire.

La méthode tabou est une procédure itérative qui, partant d'une solution initiale, tente de converger vers la solution optimale en exécutant, à chaque pas, un mouvement dans l'espace de recherche. Chaque pas consiste d'abord à engendra un ensemble de solutions voisines de la solution courante pour ensuite en choisir la meilleure, même si ce choix entraîne une augmentation de la fonction objectif à minimiser.

En acceptant de détériorer la valeur de la solution courante, le minimum local peut être évité mais, en contre partie, des parcours répétitifs sont déplorés.

Aussi, pour palier à l'inconvénient majeur des méthodes de recherche locale, la recherche Tabou a pour but d'améliorer à chaque étape, la valeur de la fonction objectif, en utilisant une mémoire afin de conserver les informations sur les solutions déjà visitées.

Cette mémoire constitue la liste Tabou qui va servir à interdire l'accès aux dernières solutions visitées. Lorsqu'un optimum local est atteint, il y a interdiction de revenir sur le même chemin.

-Un critère d'aspiration, est également utilisé pour lever l'interdiction d'utilisation d'un mouvement si ce dernier conduit à une meilleure solution.

Plusieurs stratégies ont été proposées récemment afin d'améliorer l'efficacité de la méthode tabou. L'intensification et la diversification de la recherche constituent deux d'entre elles.

-L'intensification consiste à explorer en détails une région de l'espace de recherche jugée prometteuse. Sa mise en œuvre consiste, le plus souvent, en un élargissement temporaire du voisinage de la solution courante dans le but de visiter un ensemble de solutions partageant certaines propriétés.

- **La diversification** a pour objectif de diriger la procédure de recherche vers des régions inexplorées de l'espace de recherche. La stratégie de diversification la plus simple consiste à redémarrer périodiquement le processus de recherche à partir d'une solution, générée aléatoirement ou choisie judicieusement, dans une région non encore visitée de l'ensemble des solutions admissibles.

Les domaines d'application de la recherche tabou sont vastes et variés, ils passent de l'ordonnancement à la robotique, au problème du voyageur de commerce, à l'électronique voire même aux applications médicales.

Algorithme Recherche Tabou

Etape 01 : *initialisation*

$F(x) \leftarrow$ génère solution initiale

Liste tabou= vide

$x^* = x$

$x_{min} = x$

Etape 02 :

Répéter

$X \leftarrow$ la meilleure solution du voisinage de $F(x^*)$ excepté voisins dans Liste tabou

Si $f(x) < f(x_{min})$ alors

$x_{min} \leftarrow x$

Fin si

Mise à jour (Liste tabou)

$x^* = x$

Jusqu'à condition d'arrêt

➤ **Les algorithmes génétiques [1]**

L'algorithme génétique est une classe d'adaptation stochastique des algorithmes d'optimisation de la recherche et l'optimisation. Ils ont d'abord été utilisés par les Pays-Bas (1975). L'idée de base est d'essayer d'imiter une simple image de la sélection naturelle, en vue de trouver un bon algorithme.

La première étape consiste à muter, ou varier de façon aléatoire, une collecte de l'échantillon programs.

La deuxième étape est une étape de sélection, qui est souvent fait par le biais de la mesure contre une fonction de remise en forme. Le processus est répété jusqu'à ce qu'une solution appropriée soit trouvée.

Il existe un grand nombre de différents types d'algorithmes génétiques. L'étape de mutation dépend de la façon dont l'échantillon programs sont représentés, ainsi que de savoir si le programmeur comprend diverses techniques de liaison.

Le test d'aptitude est également en hausse pour le programmeur. Comme un gradient de flux optimisation, il est possible pour que le processus reste bloqué dans un maximum local de la fonction de remise en forme.

L'un des avantage d'un algorithme génétique est qu'il ne nécessite pas de fonction de l'aptitude à être très bon, car une recherche aléatoire est effectué au lieu de la suivre la voie de la moindre résistance .Mais pour réussir, il faut qu'il y'ait de belles relations entre les paramètres modifiables à la remise en forme.

➤ **Méthode gloutonne [4]**

Une méthode gloutonne consiste à fixer à chaque étape la valeur d'une variable sans remettre en cause les choix effectués précédemment. Par exemple, une heuristique gloutonne bien connue pour la coloration introduite par Brélaz est la suivante : pour choisir le nœud suivant à colorier, prendre celui dont les nœuds adjacents sont déjà coloriés avec le plus grand nombre de couleurs différentes et lui assigner la couleur autorisée de plus petit rang possible.

Les méthodes gloutonnes sont généralement rapides, mais fournissent le plus souvent des solutions de qualité médiocre. Elles ne garantissent l'optimum que dans des cas particuliers.

Conclusion

Dans ce chapitre nous avons présenté l'optimisation combinatoire d'une manière globale. On a déduit encore que les problèmes combinatoires sont généralement

NP-difficiles ou NP-complets.

Ensuite nous avons donné 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.

Étant donné l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées.

Méthodes exactes : programmation linéaire, programmation dynamique.

Méthodes approchées : Heuristique, Méta-heuristique, Recherche tabou, Algorithmes génétiques, Méthode gloutonne.

Chapitre II :

Le problème de Bin Packing

Introduction

Le problème de Bin Packing relève de la recherche opérationnelle et de l'optimisation combinatoire. Il s'agit de déterminer le nombre minimum de conteneurs (bins) identiques nécessaires pour ranger un ensemble d'objets de tailles différentes. Il a fait l'objet depuis quelques années d'une attention croissante pour deux raisons : outre son intérêt théorique, il a de nombreuses applications dans le domaine industriel de la logistique, de l'informatique et même de l'édition.

On le retrouve notamment dans l'industrie du tissu, du métal, mais aussi dans le cadre de la découpe de bois, de verre ou même de placement des spots publicitaire dans les journaux. Le problème classique se définit en une dimension, mais il existe de nombreuses variantes en deux ou en trois dimensions.

II.1. Définition du problème de Bin Packing

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 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).

II.2 Application du problème de Bin Packing

Le problème de Bin Packing est appliqué souvent dans la fabrication. Cette partie ci-dessous cite quelques exemples d'application de ce problème :

- **Fabrication des tubes**

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.

- **Emballage des meubles**

L'emballage des 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.

- **Réalisations des tâches**

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.

- **Plomberie**

On a besoin des tuyaux de tailles différentes (au plus 5 mètres) pour un appartement. On devrait couper à partir des tuyaux de tailles de 5 mètres disponibles dans les magasins. On peut appliquer le problème de Bin Packing pour calculer combien de tuyaux minimum qu'il faut acheter.

II.3 Classifications et contraintes pratiques

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

- 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 : un 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 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.

II.4. Le problème de Bin Packing uni-, bi- et tridimensionnel

II.4.1. Bin Packing unidimensionnel 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.

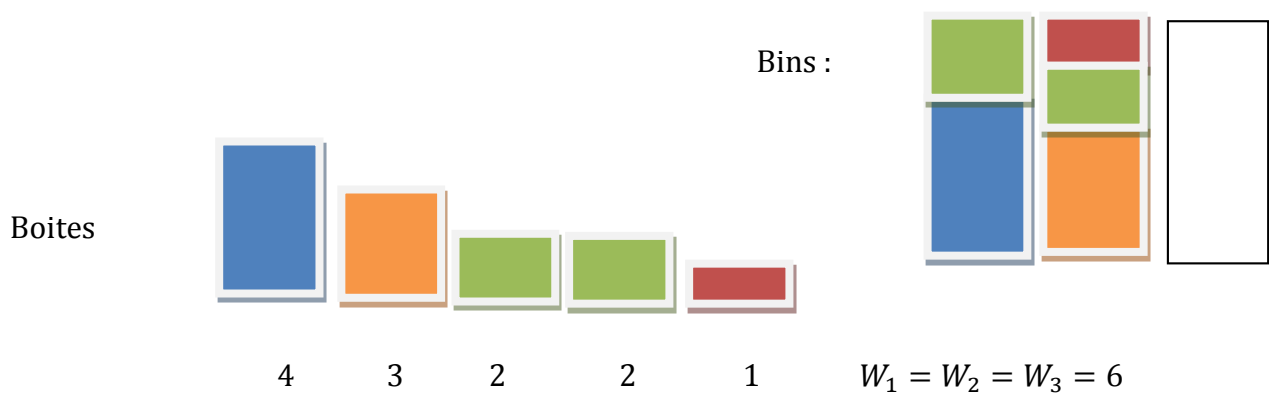
Exemple : le rangement de fichiers sur un support informatique, la découpe de câbles, le remplissage de camions ou de containers avec comme seule contrainte le poids ou le volume des articles.

Ainsi le problème de Bin Packing unidimensionnel (BPP-1D) consiste à minimiser le nombre de bin unidimensionnel 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_j longueur des bins

Objectif : minimiser le nombre de bins utilisés

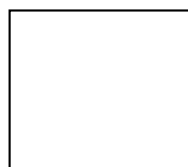
Contraintes: ne pas dépasser la capacité d'un bin.

II.4.2. Bin Packing bidimensionnel 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.

On peut dire que le problème de Bin Packing bidimensionnel (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 bin. On note (I, w, h, W, H) une instance de BPP-2D, $I = \{1, 2, \dots, n\}$ 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]

Exemple

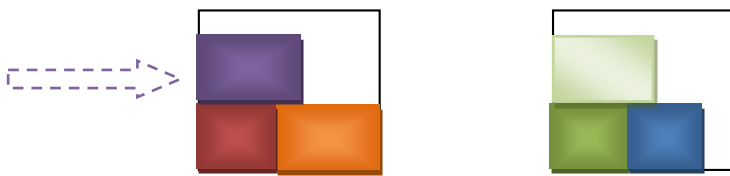
On a un ensemble infini de bins rectangulaires $W=10, H=10$



Et on un ensemble fini de nombre d'objets :

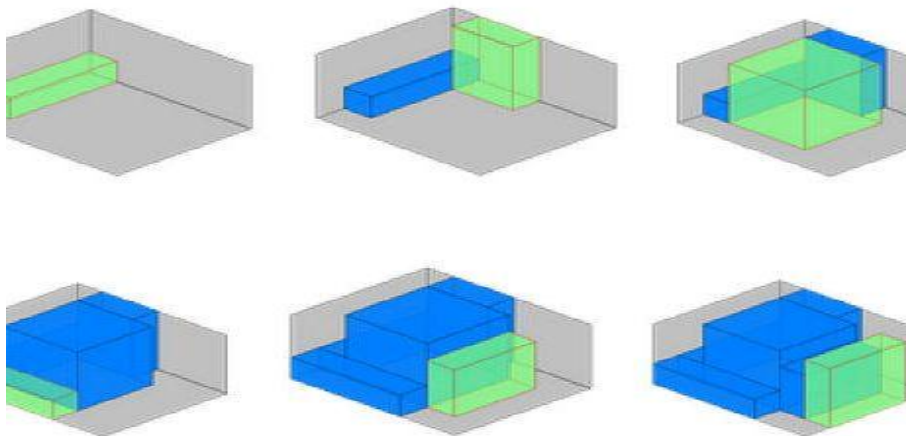


Solution:



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.

Exemple



II.4.3 Le problème de Bin Packing tridimensionnels où les objets sont rangés dans un espace de volume fixe comme le rangement d'objets dans un entrepôt, des camions, des conteneurs, etc.

Le problème de Bin Packing tridimensionnels (BPP-3D) connu aussi sous le nom de problème de chargement de conteneurs, est un problème combinatoire dont dérive de nombreuses applications industrielles. 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.

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

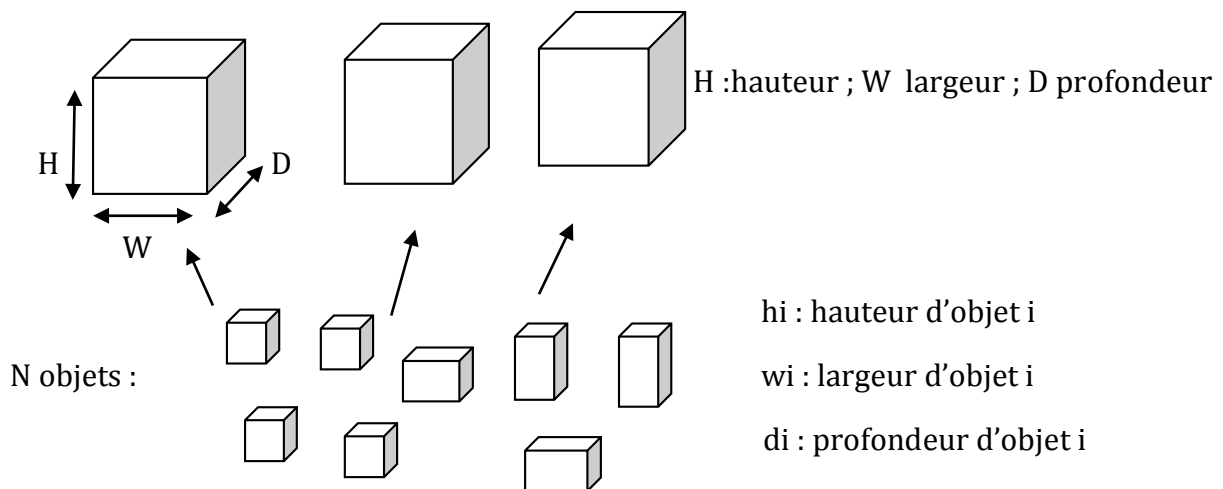


Figure : Illustration graphique du BPP-3D

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

- 1- Eviter le débordement,
- 2- Respecter le tonnage (volume intérieur) du support,
- 3- 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.

Conclusion

Dans ce chapitre nous avons définie c'est quoi le problème de Bin Packing de manière générale et nous avons commencé à donner quelques domaines d'applications de ce problème comme dans la fabrication des tubes, emballage des meubles, plomberie et la réalisation des tâches.

Ensuite nous avons classifié le problème de Bin Packing au trois types :

- Bin packing unidimensionnel (BPP-1D)
- Bin packing bidimensionnel (BPP-2D)
- Bin packing tridimensionnel (BPP-3D)

Chapitre III :
Généralité sur le problème de
placement

Introduction

Le problème de placement est un problème d'optimisation combinatoire d'intérêt majeur qui intervient dans des problématiques diverses. Le problème concret de placement se pose lorsque l'on cherche à remplir une ou plusieurs boîtes avec un ensemble fini d'objets ou lorsque l'on cherche à obtenir un ensemble fini d'objets en découpant un ou plusieurs objets de taille supérieurs et cela de la manière la plus économique possible. Dans le premier cas, il s'agit d'un problème de remplissage, et dans le deuxième cas il s'agit d'un problème de découpe.

III.1 Formulation mathématique d'un problème de placement

Le problème de placement est défini de la façon suivante : étant donné un ensemble de n objets, $Y = Y_1, \dots, Y_n$ et un nombre illimité de boîtes (rectangles identiques), de dimensions plus larges que celles des objets. Le problème consiste à déterminer le nombre minimum de boîtes à utiliser pour ranger l'ensemble de tous les objets sans chevauchement.

III.2 Modèle mathématique

Le problème de placement peut être modélisé de la manière suivante :

- un ensemble de n' d'objets qu'on appelle boîtes.

$$B = B_1, B_2, \dots, B_j, \dots, B_{n'} .$$

- un ensemble de n d'objets $Y = Y_1, Y_2, \dots, Y_i, \dots, Y_n$.

- W_j : La longueur de la boîte B_j .

- w_i : La longueur de l'objet Y_i .

$$x_{ij} = \begin{cases} 1 & \text{si l'objet } Y_i \text{ est rangé dans la boîte } B_j \\ 0 & \text{si l'objet } Y_i \text{ ne peut être rangé dans la boîte } B_j \end{cases}$$

$$B_j = \begin{cases} 1 & \text{si la boîte } B_j \text{ est utilisée} \\ 0 & \text{si la boîte } B_j \text{ n'est pas utilisée} \end{cases}$$

On cherche à minimiser le nombre de boîtes à utiliser, c'est-à-dire :

$$\text{Min } \sum_{j=1}^{n'} B_j$$

$$B_j \in \{0,1\}, j = 1, \dots, n'$$

On suppose que les Boîtes ont la même taille, c'est-à-dire :

$$\forall j, k \in \mathbb{N}, j \neq k \quad w_j = w_k$$

• **Contraintes du problème**

1. Un Objet est placé uniquement dans une seule Boîte

$$\left\{ \begin{array}{l} \sum_{j=1}^{n'} x_{ij} = 1 \\ x_{ij} \in \{0,1\} \\ i = 1, \dots, n \end{array} \right.$$

2. La longueur de l'ensemble des objets rangés dans une boîte B_j ne doit pas dépasser la longueur de ceci.

$$\left\{ \begin{array}{l} \sum_{i=1}^n w_i * x_{ij} \leq W_j \\ \forall i = 1..n, \forall j = 1..n' \end{array} \right.$$

L'inégalité signifie qu'on ne peut dépasser la taille d'une boîte pour un rangement.

A noter que la partie droite de l'inégalité oblige B_j à prendre la valeur 1 dès qu'un article est rangé dans la boîte j.

Le modèle est donc le suivant :

$$\left\{ \begin{array}{l} \text{Min } \sum_{j=1}^{n'} B_j \\ \sum_{j=1}^{n'} x_{ij} = 1 \\ \sum_{i=1}^n w_i * x_{ij} \leq W_j \\ x_{ij} \in \{0,1\} \\ i = 1, \dots, n ; j = 1, \dots, n' \end{array} \right.$$

La modélisation décrite plus haut a été proposée par Leonid Kantorovitch en 1960. Il existe d'autres formulations linéaires pour ce problème, sous forme d'un problème de flot maximum dans un graphe, ou utilisant une décomposition de Dantzig Wolfe.

Exemple

Supposons que nous avons des récipients de taille 10. Quel nombre d'entre eux sont nécessaires pour ranger les articles de taille 3, 6, 2, 1, 5, 7, 2, 4, 1, 9 ?

Une solution est d'utiliser 4 boîtes et arranger les objets comme suit

- Première boîte : on met les articles qu'ont la taille 6,4
- Deuxième boîte : on met les articles qu'ont la taille 1,9
- Troisième boîte : on met les articles qu'ont la taille 3,7
- Quatrième boîte : on met les articles qu'ont la taille 2, 5, 2,1

III.3 Classification des problèmes de placement [9]

Il y a trois grandes familles de problèmes selon le nombre de dimensions des objets :

III.3.1 problème de placement en une dimension

Le placement en une dimension est la version standard des problèmes de placement, il consiste le rangement un ensemble d'objets caractérisés par une seule variable (hauteur, largeur, poids, ou autre) dans un ensemble de boîtes de taille fixé W sans chevauchement.

Une instance de placement 1D notée D , et une paire (Y, W) où $Y = \{Y_1, \dots, Y_n\}$ est une liste des articles à ranger et W une valeur positive.

Chaque article Y_i possède une longueur w_i inférieure à W .

La valeur optimale du nombre de boîtes nécessaires pour ranger tous les objets d'une instance D est notée $OPT(D)$.

Le cas le plus fréquent consiste à utiliser des barres de métal existant dans une entreprise pour satisfaire des demandes en barres différentes. On cherche à minimiser la longueur des chutes non réutilisables (inférieure à une longueur minimale).

III.3.2 problème de placement en deux dimensions

Plus formellement, le problème de placement en deux dimensions est défini de la façon suivante : étant donné un ensemble de n objets rectangulaires $Y = \{Y_1, \dots, Y_n\}$ et un nombre illimité de rectangles identiques (boîtes) de dimensions plus larges que celles des objets, les (w_i, h_i) les dimensions d'un objet, et (W, H) les dimensions de boîte $B : B = (W, H)$. Nous considérons, sans perte de généralité, que les dimensions des objets et des boîtes sont des entiers. Une instance de placement 2D notée I est alors définie par le triplet (Y, W, H) . La valeur optimale du nombre de boîtes nécessaires pour ranger tous les objets d'une instance I est notée $OPT(I)$.

Ces problèmes se posent en particulier chez les fabricants de verre, de tôles, et chez les fabricants de vêtements. Il s'agit de placer des commandes rectangulaires sur des formes rectangulaires de formats standards en minimisant le nombre de formes à découper ou la surface des chutes non réutilisables obtenues.

III.3.3 Problème de placement à trois dimensions :

le problème de placement en trois dimensions est défini de la façon suivante : étant donné un ensemble de n objets cubiques $Y = \{Y_1, \dots, Y_n\}$ et un nombre illimité de boîtes identiques de dimensions plus larges que celles des objets, les (w_i, h_i, l_i) les dimensions d'un objet, et (W, H, L) les dimensions de boîte B . On considère que les dimensions des objets et des boîtes sont des entiers.

III.4 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 [5].

• III.4.1. 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.

• III.4.2. 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 palettes, bien que dans les applications pratiques, les deux variantes du problème de chargement puissent être distinguées par leurs contraintes.

Conclusion

Dans ce chapitre en premier nous avons modélisé le problème de placement par la formulation mathématique. Secondement on l'a classifié en trois grandes familles selon le nombre de dimensions des objets :

- Problème de placement en une dimension
- Problème de placement en deux dimensions
- Problème de placement à trois dimensions

Ensuite nous avons cité les différents types de problème de placement :

- Problème de stock de découpe
- Problème de chargement

Chapitre IV :

Problème de chargement de conteneurs

Introduction

Ce chapitre traitera les problèmes du chargement de conteneurs où nous porterons une attention particulière à cette question afin d'évaluer la pertinence pratique des recherches qui ont été menées dans le champ.

Dans le problème du chargement des conteneurs, différents types de petites boîtes rectangulaires tridimensionnelles doivent être chargées dans un conteneur rectangulaire tridimensionnel. Les boîtes doivent être emballées orthogonalement dans le conteneur, c'est-à-dire que les bords des boîtes doivent être parallèles à un bord du conteneur. Chaque boîte doit être complètement emballée dans le conteneur et ne peut pas se chevaucher avec une autre boîte.

Les boîtes ayant les mêmes dimensions et les mêmes caractéristiques, comme le poids sont regroupées dans le même type de boîte. Les boîtes sont autorisées à être accordées dans tout ou partie des six orientations possibles.

Il y a plus de vingt ans, Bischoff et Ratcliff (1995) ont soutenu : « ... que les approches aux problèmes de chargement des conteneurs ne s'appliquent chacune qu'à une partie étroite de l'éventail de situations rencontrées dans la pratique... ». Ils ont en outre affirmé « ... qu'un nombre de facteurs qui sont souvent importants dans des situations pratiques n'ont pas a reçu suffisamment d'attention dans la littérature ».

En général l'objectif est de maximiser l'utilisation du volume de conteneur ou de minimiser le volume du conteneur inutilisé. Le deuxième problème est de charger l'intégralité de la cargaison dans un ou plusieurs conteneurs. L'objectif est de minimiser le nombre de conteneurs utilisés.

IV.1 Définition et catégories

Définition

Les problèmes de chargement de conteneurs peuvent être interprétés comme des problèmes d'affectation géométrique. Ce problème est défini avec les petits objets en trois dimensions (appelés *fret*) qui doivent être affectés (emballés) dans des grands objets rectangulaires (cubiques) en trois dimensions (appelés *conteneurs*).

L'objectif est que la fonction soit optimisée et doit respecter les deux conditions de faisabilité géométrique suivantes :

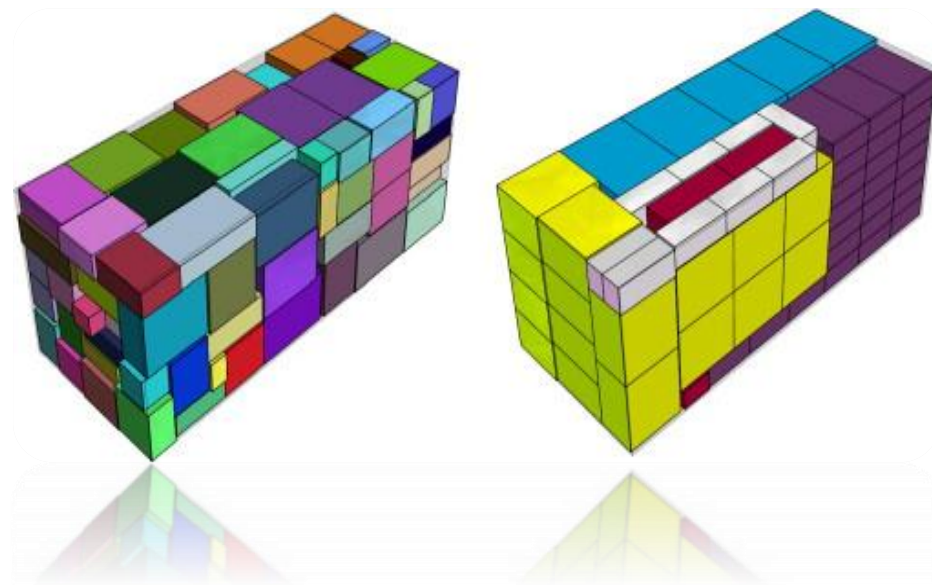
- Tous les petits objets se trouvent entièrement dans le conteneur.
- Les petits objets ne se chevauchent pas.

Une description formelle d'une solution à un problème d'affectation de ce type sera appelée « *modèle de chargement* ».

Comme un grand objet peut en fait être un vrai conteneur, mais selon la définition donnée il peut également s'agir de l'espace de chargement d'un camion ou d'une palette chargé jusqu'à une certaine hauteur.

Selon la typologie introduite par Wäscher, Haußner & Schumann, on déduit qu'il existe deux types de problème de chargement de conteneurs :

- Les problèmes de chargement de conteneurs, dans lesquels suffisamment de conteneurs sont disponibles pour accueillir tous les petits articles.
- Les problèmes de chargement de conteneurs, dans lesquels seul un sous-ensemble des articles peuvent être emballés car la disponibilité des conteneurs est limitée.



Catégories

➤ Catégorie 01

Les problèmes de « minimisation d'entrée (valeur) » sont les problèmes de chargement de conteneurs, dans lesquels suffisamment de conteneurs sont disponibles pour accueillir tous les petits articles. On va citer quelques problèmes de ce type :

- **Problème de stock : taille de stock unique (SSCSP)**

Emballage d'un ensemble de marchandises ayant des dimensions différentes dans des conteneurs identiques.

- **Problème de stock : taille de stock multiple (MSSCSP)**

Emballage d'un ensemble de marchandises ayant des dimensions différentes dans des conteneurs, de sorte que le nombre des conteneurs utilisés soit minimisée.

- **Problème d'emballage de bacs de taille unique (SBSBPP)**

Emballage d'un ensemble de marchandises ayant des dimensions différentes dans des conteneurs identiques.

- **Problème d'emballage des bacs résiduels (RBPP)**

Emballage d'un ensemble de marchandises ayant des dimensions différentes dans des différents conteneurs de sorte que le nombre des conteneurs utilisés soit minimisée.

- **Problème de dimension ouverte (ODP)**

Emballage d'un ensemble de marchandises dans un seul conteneur avec une ou plusieurs dimensions variables de telle sorte que le volume du conteneur soit minimisé.

➤ Catégorie 02

Les problèmes de « maximisation de sortie (valeur) » sont les problèmes de chargement de conteneurs, dans lesquels seul un sous-ensemble des articles peuvent être emballés car la disponibilité des conteneurs est limitée. On peut citer quelques problèmes :

- **Problème d'emballage d'articles identiques (IIPP)**

Chargement un seul conteneur avec un nombre maximum de petits articles identiques.

- **Problème de placement de plusieurs gros objets identiques (MILOPP)**

Chargement d'un ensemble de conteneurs identiques avec une sélection de marchandises tel que la valeur des marchandises chargés soit maximisée.

• **Problème de placement hétérogène de grands objets (MHLOPP)**

Chargement d'un ensemble de conteneurs ayant des dimensions différentes avec une sélection de marchandises de sorte que la valeur des articles chargés est maximisée.

• **Problème de sac à dos unique (SKP)**

Chargement d'un seul conteneur avec une sélection de cargaison telle que la valeur des cargaisons chargés soit maximisée.

• **Problème de sac à dos identique et multiple (MIKP)**

Chargement d'un ensemble de conteneurs identiques avec une sélection de marchandises tel que la valeur des marchandises chargés soit maximisée.

On note que *la maximisation de la sortie (valeur)* est équivalente à la maximisation d'utilisation du volume de conteneur si la valeur des petits articles est proportionnelle à leur volume.

La présentation suivante sera basée sur cette catégorisation. Nous notons qu'en ce qui concerne la définition des termes "fret" et "conteneur" donné ci-dessus et les problèmes de chargement des conteneurs sont considérés ici comme des problèmes tridimensionnels (3D) ou des problèmes de découpe et d'emballage (C&P).

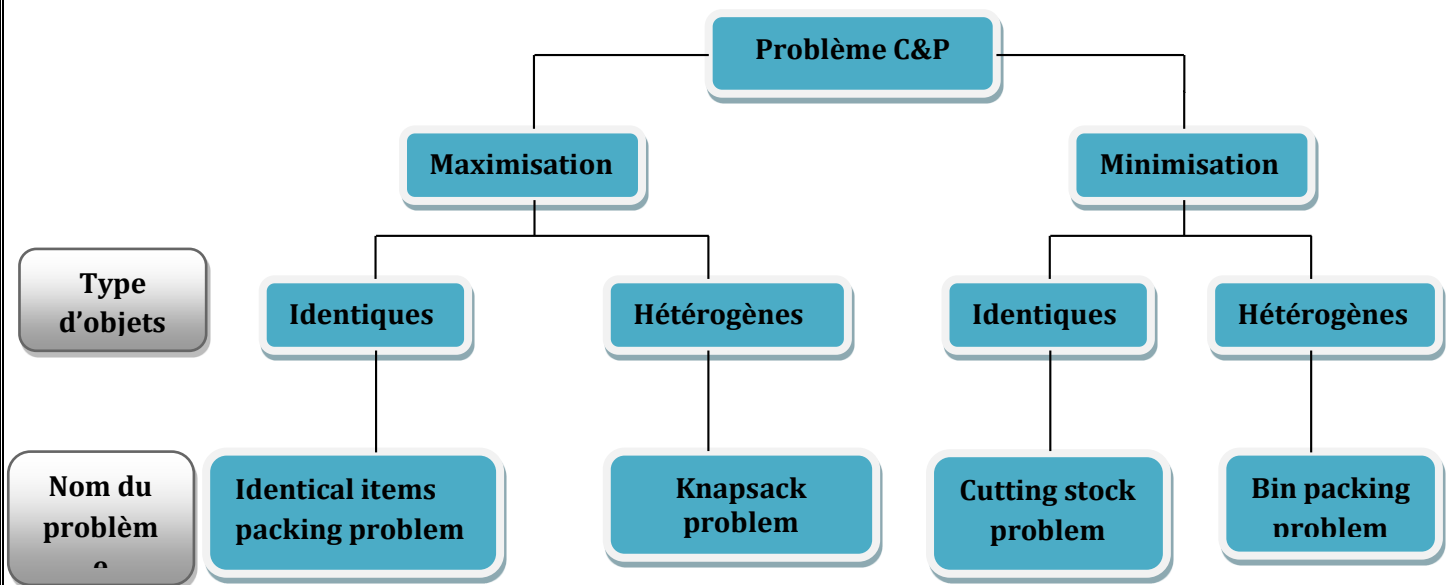


Figure : les différents types de problème de découpe et d'emballage (C&P)

En général, les petits objets peuvent avoir une forme régulière (rectangulaire, sphérique...) ou forme irrégulière. Cependant, à quelques exceptions près, les publications dans le domaine de chargement des conteneurs uniquement avec de petits objets rectangulaires. Comme c'est l'usage linguistique général dans la littérature, nous qualifierons ces éléments de « boîtes ».

En outre, se référant aux problèmes standard de découpe et d'emballage (C&P), il est impliqué que certaines hypothèses (par exemple concernant la fonction objective, les assortiments de boîtes et conteneurs, etc.) tiennent pour les problèmes de chargement des conteneurs. En particulier, chaque fois que des boîtes doivent être chargées il faut que leurs bords seront orthogonaux, c'est-à-dire les surfaces des boîtes doivent être alignées parallèlement au sol et aux parois du conteneur (voir la figure au-dessous)

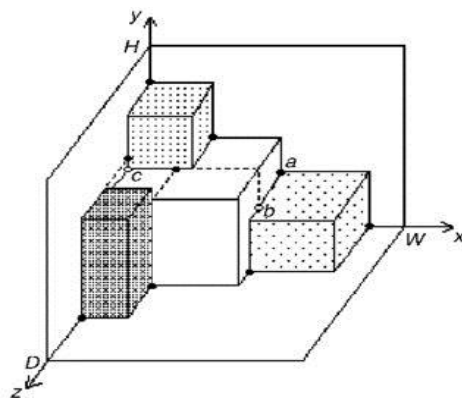


Figure : Placement des boîtes dans un conteneur.

IV.2. Revue de la littérature

Sur la base du nombre de types de boîtes différents, le problème de chargement des conteneurs peut être classé en trois types :

- Le problème homogène avec un seul type de boîte.
- Le problème faiblement hétérogène avec relativement peu de types de boîtes différentes.
- Le problème fortement hétérogène avec un grand nombre de types de boîtes différentes.

Le problème de chargement des conteneurs avec des boîtes homogènes s'est réduit au « problème de chargement des palettes » en deux dimensions (PLP) en supposant que les boîtes ne peuvent être placées que verticalement. Nelißen [10] a fourni un excellent aperçu de ce problème.

Les algorithmes pour les problèmes de chargement de conteneurs simples et multiples utilisent l'heuristique Gourmande (greedy) [11, 12] pour le problème de chargement des palettes (PLP) en tant que sous-routine pour emballer une certaine quantité de boîtes de type unique sur une surface donnée.

Pour le problème de chargement de conteneurs hétérogènes, plus d'un type de boîte doit être chargé dans le conteneur. En raison de la complexité du problème, la plupart des résultats sur ce sujet sont basés sur les heuristique et les méta-heuristiques.

Alors il existe deux approches de base utilisées dans l'heuristique pour résoudre le problème de chargement d'un conteneur unique avec des boîtes hétérogènes [13]. Ces méthodes sont basées sur la construction de murs et la construction de couches. L'approche de construction de murs construit des murs verticaux sur toute la longueur ou la largeur du conteneur, tandis que l'approche de construction de couches construit le plan de chargement couche par couche à partir du plancher du conteneur vers le haut.

IV.3 Contraintes dans le chargement des conteneurs

Dans cette section, nous présenterons les contraintes pratiquement pertinentes qui peuvent être rencontrées dans les problèmes de chargement des conteneurs. Nous distinguerons les contraintes liées aux grands objets (contraintes liées aux conteneurs) et celles liées aux petits objets, où ces derniers peuvent faire référence à un élément individuel (contraintes liées à l'élément) ou à l'ensemble ou à un sous-ensemble d'articles (contraintes liées au fret). De plus, les contraintes peuvent être liées à la relation entre les gros objets et les petits objets. Ils se manifestent dans les positionnements des contraintes des petits objets dans les conteneurs. Enfin, les contraintes peuvent être liées au résultat du processus d'emballage, c'est-à-dire de la charge (contraintes liées à la charge).

Les contraintes dans le chargement des conteneurs peuvent se produire sous forme de contraintes dures ou douces.

Contraintes dures : doit être satisfait un schéma de chargement qui viole une contrainte dure.

Contraintes douces ne devraient tenir qu'au moins dans certaines limites et les violations sont tolérées.

➤ IV.3.1 Contraintes liées aux conteneurs

IV.3.1.1 Limites de poids

En règle générale, un conteneur ne peut être chargé qu'avec de petits articles tant qu'une certaine limite de poids n'est pas dépassée. Ces contraintes peuvent ne pas toujours apparaître (c.-à-d. Lorsque l'ensemble des petits articles à emballer sont constitués de meubles en caoutchouc mousse), mais ils le feront chaque fois que des objets lourds doivent être chargés. Dans de tels cas, les limites de poids peuvent sembler être plus restrictive que les contraintes d'espace imposées par les dimensions des conteneurs.

Les limites de poids peuvent être modélisées de manière simple en tant que contraintes linéaires de sac à dos, où la somme des poids des articles chargés doit être

inférieure ou égale à la limite de poids imposée par le conteneur. Dans les algorithmes de chargement des conteneurs, ils permettent une vérification simple et rapide de la faisabilité des solutions. Il convient de noter que les contraintes de poids reflètent un problème régulièrement rencontré des publications qui visent une solution intégrative d'une extension particulière d'un problème *tridimensionnel* (3D) de découpe et d'emballage « C&P ».

Comme Dereli & Das (2010) traitent d'un problème de type SKP dans lequel la capacité de chargement de conteneur est limitée en volume et en poids. Cependant, outre le ciblage maximal utilisation du volume du conteneur uniquement, ils essaient également de charger le conteneur avec une cargaison de poids. Cela conduit à un problème d'optimisation avec deux fonctions objectives, pour lesquelles ils donner une formulation de programmation d'objectifs.

IV.3.1.2 Contraintes de répartition du poids

Contraintes de répartition du poids (aussi: *contraintes d'équilibrage de charge*) exigent que le poids de la cargaison soit réparti aussi uniformément possible à travers le plancher du conteneur. Des charges équilibrées réduisent le risque de déplacement de la cargaison pendant que le conteneur est déplacé. Des charges déséquilibrées peuvent entraîner des irrégularités inacceptables répartition des poids à l'essieu lorsque le conteneur est transporté sur un camion. Certaines opérations de manutention (par exemple les opérations de levage appliquées au conteneur) peuvent même être complètement impossible.

Afin de parvenir à une répartition du poids, on peut exiger que le centre de gravité de la charge soit proche du milieu géométrique du fond du conteneur ou ne doit pas dépasser une certaine distance.

En d'autres termes, les contraintes de distribution de poids représentent des contraintes souples. C'est pour cela Chen, Lee et Shen [14] (1995) introduire un modèle d'optimisation d'entiers mixtes (linéaire) pour tous les types d'entrée problématiques comme la minimisation et la maximisation de la production et montrer comment la contrainte d'équilibrage de charge dimensionnelle peut être intégrée dans ce modèle.

➤ IV.3.2. Contraintes liées aux articles

IV.3.2.1 Priorités de chargement

Le type de contrainte abordé dans cette section ne peut survenir qu'en conjonction avec le problème de chargement de type de *maximisation de sortie (valeur)*. L'espace conteneur disponible étant pas suffisant pour accueillir tous les petits articles.

Alors il faut décider quels articles doivent être chargés et lesquels doivent être laissés. En pratique, le chargement de certains articles peut être plus souhaitable que le chargement des autres, c'est-à-dire les priorités de chargement existent pour les articles. De telles

priorités peuvent résulter, par exemple, des délais de livraison ou des exigences liées à la fraîcheur ou à la durée de conservation des produits.

En règle générale, un sous-ensemble d'éléments doit être chargé, ce qui entraîne des contraintes strictes. Ces derniers éléments peuvent être différenciés davantage en classes de priorités différentes. Une telle priorité peut refléter une condition dans laquelle aucun article de priorité inférieure ne doit être expédié s'il nécessite un article d'une priorité plus élevée à laisser (priorités absolues), ou ils peuvent simplement représenter la valeur de placer un article dans un conteneur au lieu d'un autre (priorités relatives).

Même si les priorités de chargement sont parfois caractérisées comme des contraintes importantes (exemple : Junqueira), ils ne sont presque jamais explicitement pris en compte dans la conception d'algorithmes de chargement de conteneurs, et même Ren, Tian et Sawaragi présentent un algorithme pour problèmes des types SLOPP et SKP ou les problèmes qui ont des éléments d'une priorité faible et élevée (absolue). L'utilisation du volume des conteneurs doit être maximisée sous la contrainte supplémentaire tous les éléments de haute priorité sont chargés, c'est-à-dire que les priorités de chargement dur sont prises en compte.

Bortfeldt & Gehring (1999) proposent un algorithme génétique pour les problèmes du type SKP (sac à dos) et SLOPP (Problème de placement d'un seul grand objet) qui permet des priorités de chargement doux et dur.

Il y a deux classes de priorités, c'est-à-dire que chaque case a une priorité élevée ou faible. Si des priorités de chargement dur existent pour certaines boîtes, les boîtes de faible priorité ne doivent pas être incluses dans le (s) schéma (s) de chargement, sauf si toutes des boîtes de haute priorité ont été installées.

Priorités douces, c'est-à-dire l'inclusion des cases prioritaires du schéma de chargement sont gérées par la fonction objective.

IV.3.2.2 Contraintes d'orientation

En principe, chaque dimension d'une boîte peut servir de hauteur, donnant lieu à trois orientations verticales. En sélectionnant une dimension particulière comme hauteur, l'orientation *verticale* est définie. Puis, étant donné que seuls les modèles de chargement orthogonaux sont autorisés, la boîte peut être alignée horizontalement sur les parois du conteneur au moyen de deux *orientations horizontales*.

En d'autres termes: ($3 \times 2 = 6$) il existe six orientations selon lesquelles une boîte (rectangulaire) peut être placée orthogonalement dans un récipient.

En pratique, le nombre admissible d'orientation d'une boîte peuvent être limitées à la fois verticalement et horizontalement.

Les contraintes d'orientation limitent généralement l'orientation verticale d'une boîte à une dimension ou à deux dimensions (par exemple, dans le cas de longues mais basses et étroites boîtes qui ne doivent pas être placées sur sa plus petite surface). Aussi la force

portante d'une boîte dépend de son orientation verticale. Par conséquent, toutes les orientations verticales peuvent être utilisées lors du chargement d'un conteneur. Il est même possible qu'une orientation particulière est possible sur un niveau de charge plus élevé (couche) qui n'est pas autorisé sur un inférieur.

Des contraintes d'orientation verticale sont introduites afin d'empêcher l'emballage d'être endommagé ou afin d'assurer la stabilité de la charge. Outre ces contraintes qui limitent l'orientation verticale d'une boîte, des contraintes peut être actif, ce qui restreint *l'orientation horizontale* d'une boîte.

Comme Bischoff & Ratcliff mentionnent qu'une palette (entrée bidirectionnelle) qui doit être chargée par un camion chariot élévateur ne peut être approché que des deux côtés, « l'avant » et « l'arrière ».

Les contraintes d'orientations représentent le type de contrainte le plus fréquemment traité. Alors on peut distinguer (qui incluent le cas sans contrainte) cinq (5) cas possibles :

Cas 1: Une seule orientation est autorisée pour chaque boîte (type) à la fois verticale et direction horizontale, c'est-à-dire que les boîtes ne peuvent pas être tournées. Cette hypothèse peut être liée, par exemple, aux cas où un conteneur doit être chargé de palettes (en trois dimensions) qui ne peuvent être approché par un chariot élévateur d'un côté particulier (et de son côté opposé).

Cas 2: Une seule orientation verticale est autorisée pour chaque boîte (type) alors qu'aucune restriction n'est donnée en ce qui concerne leur direction horizontale.

Étant donné que seuls les schémas de chargement orthogonaux sont généralement autorisés les boîtes pour être chargées dans un conteneur rectangulaire, cette contrainte permet pratiquement rotations à 90 ° des boîtes sur le plan horizontal. Une telle contrainte reflète une situation dans laquelle toutes les boîtes ne peuvent être posées que sur une surface particulière, par exemple lorsque toutes les cases sont marquées avec le signe "de cette façon!".

Cas 3: Il n'y a pas de restriction générale quant à l'orientation des boîtes dans la direction verticale. Cependant, jusqu'à deux orientations verticales peuvent être interdites pour chaque boîte (type). Dans le sens horizontal, l'orientation est libre, mais en raison de la limitation à motifs orthogonaux les boîtes peuvent être tournées par pas de 90 °.

Cas 4: Il n'y a pas de restriction générale quant à l'orientation des boîtes dans la direction verticale et horizontale. Cependant, jusqu'à cinq orientations peuvent être interdites pour chaque boîte (type). Ce cas comprend la plus grande variété d'orientations différentes contraintes. En plus les boîtes non rotatives (cas1) peuvent également être traitées.

Cas 5: Il n'existe aucune contrainte quant à l'orientation des cases, ni dans vertical ni horizontal. Toutes les boîtes peuvent tourner librement. Par rapport aux autres

paramètres, celui-ci garantit le plus grand degré de liberté, c'est-à-dire le plus grand espace de solution.

Les contraintes d'orientation verticale et horizontale sont traitées comme des contraintes dures.

IV.3.2.3 Contraintes d'empilement

Contraintes d'empilement (aussi: *contraintes portantes*) restreignent la façon dont les boîtes peuvent être placées les unes sur les autres. Ils résultent du peu *résistance portante* des caisses. Combien de poids ou de pression peut supporter une boîte avant il va éclater dépend en premier lieu de la force du boîtier, qui est déterminé par la construction du boîtier et le matériau utilisé.

Cependant, la charge, la force portante ne peut pas nécessairement être simplement mesurée par le poids maximum qui peut être appliqué par unité de surface de la boîte de support. Au lieu de cela, la résistance portante est souvent déterminée par la résistance des parois latérales du caisson, donc l'écrasement des bords, c'est-à-dire le poids ou la masse qui écraserait le boîtier de la boîte lorsqu'elle était appliquée vers le bas sur son bord, peut-être une mesure plus appropriée.

Comme cela a déjà été mentionné, une boîte peut avoir plusieurs orientations verticales admissibles, et sa résistance à la charge peut varier avec les orientations dans lesquelles il est placé à l'intérieur du conteneur. Il peut être davantage affecté par d'autres facteurs comme :

- Le contenu des boîtes (exemple, des boîtes complètement remplies de solides le contenu comme le bois dur permet généralement un empilage plus élevé que les boîtes qui ne sont incomplètement remplis de contenu moins solide).
- Les conditions dans lesquelles les boîtes sont utilisé, y compris l'humidité, la durée de la charge et le mode d'empilage (colonne gerbage ou gerbage inter blocage).

Les contraintes d'empilement, là encore, sont généralement introduites afin d'éviter d'endommager les boîtes et afin de protéger les marchandises et les emballages. Les contraintes de ce type sont considérées comme des contraintes fortes. Et la cargaison peut être divisée en sous-ensembles de boîtes fragiles et non fragiles.

Les boîtes non fragiles ne peuvent être placées que sur d'autres non fragiles, mais pas sur des fragiles, tandis que les boîtes fragiles peuvent être posées sur des boîtes non fragiles et sur d'autres fragiles. Celle-là est une approche consiste à interdire de placer un type particulier de boîte "i" au-dessus.

Un autre type "j" par exemple en interdisant de placer la plus grande boîte sur des plus petites. De même, le nombre de boîtes qui peuvent être empilées les unes sur les autres

peuvent également être limitées. Les contraintes de ce dernier type peuvent, par exemple, représenter la hauteur maximale.

Instruction souvent rencontrée dans la pratique (Bischoff 2006). Lin et al. (1993) comme ainsi exigent que les objets plus lourds soient placés sous les plus légers.

Ainsi ils envisagent d'emballer des articles de densité plus élevée en dessous des éléments de densité inférieure.

Dans une approche plus générale, *la résistance portante limitée* peut être représentée par la pression maximale (**unités de poids par unité de surface, kg / m²**) qui peut être imposée particulier de boîte ou de type de boîte sans le déformer et endommager son contenu. Une mesure de ce type, cependant, implique que la même pression peut être appliquée n'importe où sur la surface supérieure de la boîte de support. Cela ne reflète pas le fait que -en raison de sa construction - la boîte peut résister à une pression plus élevée sur les bords qu'au centre de sa surface supérieure.

De plus, la rigidité de la surface supérieure d'une boîte détermine comment le poids d'une boîte qui est mis dessus est effectivement transmis. Si la surface supérieure de la boîte située en dessous est en matériau souple (comme le carton), alors le poids sera transmis - plus ou moins -directement dans la zone de contact, uniquement. Si la surface supérieure est constituée d'un matériau très rigide (comme une plaque métallique), le poids sera réparti sur toute la surface supérieure de la boîte de soutien.

➤ IV.3.3 Contraintes liées au fret

IV.3.3.1 Contraintes d'expédition complète

Les problèmes de type de maximisation de la sortie (valeur) nécessitent que la cargaison soit logée de la meilleure façon possible, mais comme il n'y a pas assez d'espace disponible, certains articles doivent inévitablement être laissés pour compte. Dans ce cas, certains sous-ensembles d'articles à charger peuvent désormais représenter des entités fonctionnelles ou administratives.

Si un élément d'un sous-ensemble est chargé, tous les autres éléments de ce sous-ensemble doivent également être chargés. Si un élément ne peut pas être chargé, aucun élément du sous-ensemble ne sera chargé du tout. Tel des contraintes d'expédition complète sont rencontrées, par exemple, lorsque des parties d'un meuble (meuble de cuisine, armoire encastrée, etc.) sont emballés séparément et doivent être assemblés sur site chez un client.

Dans ce cas, les expéditions incomplètes ne sont généralement pas autorisées. De toute évidence, les contraintes d'expédition complète n'ont de sens que pour les problèmes de sortie (valeur) type de maximisation et ils représentent généralement des contraintes dures. Deux cas peuvent être distingués:

Dans le premier cas : si au moins un article d'un sous-ensemble de fret chargé, il suffit que tous les articles soient inclus dans l'envoi, mais pas nécessairement dans le même récipient.

Dans le deuxième cas : tous les éléments d'un sous-ensemble respectif doivent être chargé dans le même conteneur.

IV.3.3.2 Contraintes d'allocation

Des contraintes d'allocation ne se posent que dans plusieurs problèmes de chargement de conteneurs. D'une part, il peut exiger que les éléments d'un sous-ensemble particulier doivent aller dans le même conteneur (contraintes *de connectivité*), par exemple lorsqu'ils doivent être expédiés à la même destination ou lorsqu'ils se rendent chez un client qui souhaite recevoir tous les articles commandés, envoi unique et non comme envoi en plusieurs parties.

En revanche, il peut exister des contraintes d'allocation qui exigent certains articles ou en cours de chargement dans le même conteneur (*contraintes de séparation*). Typiquement, la nourriture et les articles de parfumerie ne doivent pas être chargés ensemble dans un seul conteneur.

Les contraintes d'allocation sont traitées comme des contraintes souples. Et elles sont également une caractéristique standard dans les publications concernant problème de chargement des conteneurs et de routage des véhicules où ils sont considérés comme contraintes difficiles.

IV.3.3.3 Contraintes de positionnement

Les contraintes de positionnement limitent l'emplacement des éléments dans le conteneur, soit en absolu termes (c.-à-d. où les articles doivent être situés ou non dans le conteneur) ou dans termes relatifs (c'est-à-dire où les articles doivent être localisés ou non par rapport à chacun autre).

Des contraintes de positionnement absolues qui exigent que certains articles doivent être placés (ou ne pas être placés) dans une position particulière ou dans une zone du conteneur.

Ces contraintes sont généralement imposées par la taille, le poids ou le contenu d'un élément. Par exemple, les objets volumineux ne peuvent souvent être (dé) chargés que s'ils sont situés à côté d'une porte de conteneur.

Un problème de chargement de palettes en trois dimensions où le chargement est commencé avec la cueillette d'une grande boîte et la placer dans un coin de la palette.

Ainsi Nielsen & Odgaard proposent une recherche de quartier pour l'imbrication 2D et 3D problèmes où le grand objet peut être divisé en régions de qualité. L'affectation des petits objets sont confinés aux régions de qualité appropriée du grand objet afin de garantir que leurs exigences de qualité sont respectées.

Les contraintes de positionnement relatives qui exigent certains sous-ensembles d'articles soient placés étroitement ensemble dans le conteneur, ou, au moins, situés à une certaine distance les uns des autres (aussi: contraintes de regroupement). C'est généralement le cas si des sous-ensembles de fret peuvent être identifiés doivent être livrés à des clients spécifiques chacun. Placer étroitement les articles d'un client ensemble permettra - lors des opérations de chargement / déchargement- de faciliter le contrôle qui sera complet et réduit le nombre d'erreurs.

Les contraintes de positionnement relatives, d'autre part, peuvent également demander certains (sous-ensembles) être placés à côté ou à proximité les uns des autres. Encore une fois, les éléments qui affecteront la qualité de l'autre d'une manière négative (comme la nourriture et l'essence) ne doit pas être placée à côté L'une et l'autre.

Les situations de sauts multiples aboutissent à des combinaisons d'absolu et contraintes de positionnement relatives. Ils se caractérisent par le fait que des sous-ensembles des articles vont à différents clients. Les éléments de chaque sous-ensemble ne doivent pas seulement se trouver dans proximité les uns des autres, mais la disposition des sous-ensembles à l'intérieur du conteneur devrait également refléter la séquence selon laquelle ils doivent être livrés à leurs diverses destinations afin d'éviter des opérations de déchargement et de rechargement inutiles.

Dans l'algorithme proposé par Lai, Xue & Xu, le conteneur est partitionné en sections longitudinales, et la cargaison est affectée à ces sections dans l'ordre inverse de la séquence dans laquelle les clients sont visités plus tard.

Un problème de chargement de conteneurs rencontré dans les entreprises fournissant des services de livraison à domicile. Le problème considéré est de type SLOPP (Problème de placement d'un seul grand objet) et inclut une situation de multi-drop où tous les articles peuvent aller à différents clients.

Ce type de situation multi-drop pour un problème de type SKP vise à faciliter les opérations de déchargement. Ce type requiert qu'à chaque arrêt, les articles demandés soient disponibles sans réarranger les autres. Cette condition est remplie si les articles sont chargés dans le conteneur (espace de chargement de camion) selon un chargement / déchargement « *Last-in-First-out (LIFO)* ».

En particulier, si la destination d'un élément "i" doit être visité avant la destination d'un deuxième élément "j", alors "j" ne doit pas être placé au-dessus de "i" ou entre la porte du conteneur et "i". Dans ce cas, à chaque point de dépôt, les articles respectifs peuvent être retirés un par un de conteneur par une séquence de mouvements droits (un par article) vers le conteneur porte.

➤ IV.3.4 Contraintes liées à la charge

Les contraintes liées à la charge se réfèrent aux propriétés souhaitables ou nécessaires de l'agencement final des articles dans le conteneur.

IV.3.4.1 Contraintes de stabilité

La stabilité de la charge est souvent considérée comme l'un des problèmes les plus importants au-delà de l'utilisation de l'espace des conteneurs.

Les charges peuvent endommager la cargaison et même blesser le personnel pendant transport et / ou pendant les opérations de chargement et de déchargement. Malgré son importance apparente, les problèmes de stabilité de la charge ne sont souvent pas pris en compte dans le chargement des conteneurs. Les auteurs soutiennent que la stabilité devient une conséquence de la compacité de la charge correspondante lorsque l'espace du conteneur est élevé, l'utilisation peut être garantie.

Cela est généralement vrai pour les problèmes de maximisation de la sortie (valeur) type dans lequel seul un sous-ensemble des petits articles peut être emballé depuis la disponibilité de conteneurs sont limités. Pratiquement, la stabilité de la charge peut également être obtenue par des supports supplémentaires ou l'utilisation d'un matériau de remplissage (comme des morceaux de mousse) qui est introduit dans les petits espaces restants.

En ce qui concerne la stabilité de la charge, on peut distinguer entre la stabilité verticale et la stabilité horizontale.

Stabilité verticale (*aussi: stabilité statique*) : empêche les objets de tomber sur le plancher du conteneur ou sur d'autres objets. Il traite de la situation lorsque le conteneur n'est pas déplacé et décrit la capacité de la charge à supporter (la force de gravité).

Les problèmes de stabilité verticale sont généralement abordés en exigeant que la base d'une boîte doive être supportée (au total ou partiellement) soit par le plancher du conteneur, soit par un espace au même niveau de hauteur) fourni par les surfaces supérieures des autres boîtes. Le nécessaire c'est d'avoir un soutien apporté à toute la zone de base.

De Castro Silva, Soma & Maculan supposent que toutes les boîtes ont la même densité et que leurs poids sont proportionnels à leurs volumes. Ainsi, le centre de gravité et le centre géométrique de chaque boîte coïncide. Ensuite, en ce qui concerne les forces et les moments tournants, les auteurs formulent des conditions d'équilibre pour la localisation des centres géométriques qui constituent des charges verticalement stables. Leurs considérations sont intégrées dans un algorithme heuristique de résolution des problèmes de chargement de conteneurs de type SBSBPP (Problème d'emballage de bacs de taille unique).

Les auteurs démontrent qu'une charge peut être instable même si le centre de gravité de chaque boîte est pris en charge par le bas. En d'autres termes, l'assouplissement de la condition de support à 100% peut entraîner dans des modèles de chargement instables et, par conséquent, doivent être manipulés avec soin.

La stabilité horizontale (ou: *stabilité dynamique*)

La stabilité horizontale totale peut être considérée comme garantie si chaque article emballé est (horizontalement) à côté d'un autre article ou d'une paroi de conteneur. En conséquence, évaluent le support latéral d'une charge par le pourcentage d'articles qui ne sont en contact avec aucun autre article ou une paroi de récipient sur au moins trois des quatre surfaces (latérales).

Dans le chargement de palettes (en trois dimensions), la stabilité horizontale d'un chargement de palettes peut être améliorée en « emboîtant » les différentes couches de la boîte. On introduit deux critères pour mesurer le degré de verrouillage :

(i) Critère de soutien: la base de chaque boîte doit être en contact direct avec le haut surfaces d'au moins deux autres boîtes (ou la palette).

(ii) Critère de contact de base: au moins un certain pourcentage x de la surface de base de chaque boîte doit être supporté par la couche (ou la palette).

L'emboîtement des boîtiers, selon ces critères, est stimulé par des algorithmes proposé. De plus, assez fréquemment, des procédures de post-traitement spécifiques sont utilisées pour rendre les solutions précédemment générées plus compactes et stables.

On note également que les mesures de stabilité ne devraient pas seulement être appliquées à un ensemble mais doit également tenir compte de la dynamique de stabilité lors du déchargement d'un conteneur à plusieurs arrêts. Une charge initialement stable peut devenir instable après que certaines parties été déchargé.

IV.3.4.2 Contraintes de complexité

Les schémas de chargement complexes, d'une part, peuvent ne pas être acceptables pour les conteneurs manuels chargement, parce que ces modèles ne peuvent pas toujours être visualisés de manière à être correctement compris par le personnel de chargement et leur mise en œuvre peut être trop consommant. Des technologies mécaniques et automatiques de chargement / déchargement plus avancées.

D'autre part, ne conviennent pas toujours aux arrangements de cargaison complexes et peuvent nécessiter l'implication d'une main-d'œuvre supplémentaire et coûteuse. Les contraintes de complexité reflètent les limites des ressources technologiques et humaines.

Nous pouvons citer en exemple, un *modèle* pouvant être emballé par un *robot* Cette méthode qui peut être mis en œuvre successivement pour placer des boîtes, en partant du coin gauche à l'arrière du conteneur et en plaçant chaque autre boîte soit devant, à droite ou au-dessus de celles précédemment placées.

Alors ce modèle se réfère à une situation dans laquelle les boîtes sont emballées par un robot équipé d'une « main » artificielle parallèle au conteneur ou à la base de palette et qui - au moyen de cellules à vide - est capable de soulever des boîtes et de les libérer au poste désigné.

Conclusion

Nous avons présenté dans ce chapitre les deux catégories de problème de chargement de conteneur.

Ensuite nous avons cité les différentes contraintes liées aux chargement des conteneurs :

- Contraintes liées aux conteneurs
- Contraintes liées aux articles
- Contraintes liées aux fret
- Contraintes liées à la charge

Chapitre V : Application

V.1 Résolution des problèmes de chargement de conteneur unique

Dans l'algorithme pour les problèmes de chargement de conteneurs uniques, chaque type de boîte se voit attribuer un facteur de priorité. Le facteur de priorité (blâme) de chaque type de boîte est initialement un et mis à jour la dynamique de l'itération à l'itération pendant l'exécution de l'algorithme.

Pendant une itération de l'algorithme, une heuristique gourmande (greedy) intégrée à une procédure de recherche qu'est appliquée pour charger le conteneur. La meilleure solution trouvée par l'heuristique gourmande (greedy) est analysée, et les facteurs de priorité sont mis à jour avec les connaissances apprises de l'analyse. Les types de boîtes qui sont considérés comme des « fauteurs de troubles » se voient attribuer plus de blâme, c'est-à-dire leurs facteurs de priorité sont davantage augmentés.

Ensuite, l'heuristique de gourmande est à nouveau appliquée et les procédures ci-dessus sont répétées. Ce cycle “ construction / analyse / Priorisation “ continue jusqu'à ce que le nombre maximal d'itérations soit atteint ou que la limite de temps soit dépassée. Dans cette heuristique les types de boîtes avec des facteurs de priorité plus élevée seraient traités dans les premières étapes c'est-à-dire qu'ils seraient emballés sur des surfaces inférieures. Par rapport au concept de facteur de priorité, l'analyse des solutions construites par cette heuristique et la priorisation guidée par l'analyse permettent à l'algorithme d'apprendre des itérations précédentes pour mettre à jour les facteurs de priorité.

V.1.1. L'heuristique gourmande (greedy)

L'algorithme gourmande (greedy) construit le plan de chargement d'un conteneur unique couche par couche de bas en haut.

L'algorithme gourmande

- Au stade initial, la liste des surfaces disponibles ne contient que la surface initiale de taille “L x W” avec sa position initiale à hauteur 0.
- À chaque étape, l'algorithme sélectionne la surface utilisable la plus basse, puis détermine le type de boîte à emballer sur la surface.
- Le nombre de boîtes et le rectangle sont les boîtes à emballer, par la procédure « couche de sélection ».
- Calcule une couche de boîtes du même type avec la valeur d'évaluation la plus élevée.

Exemple

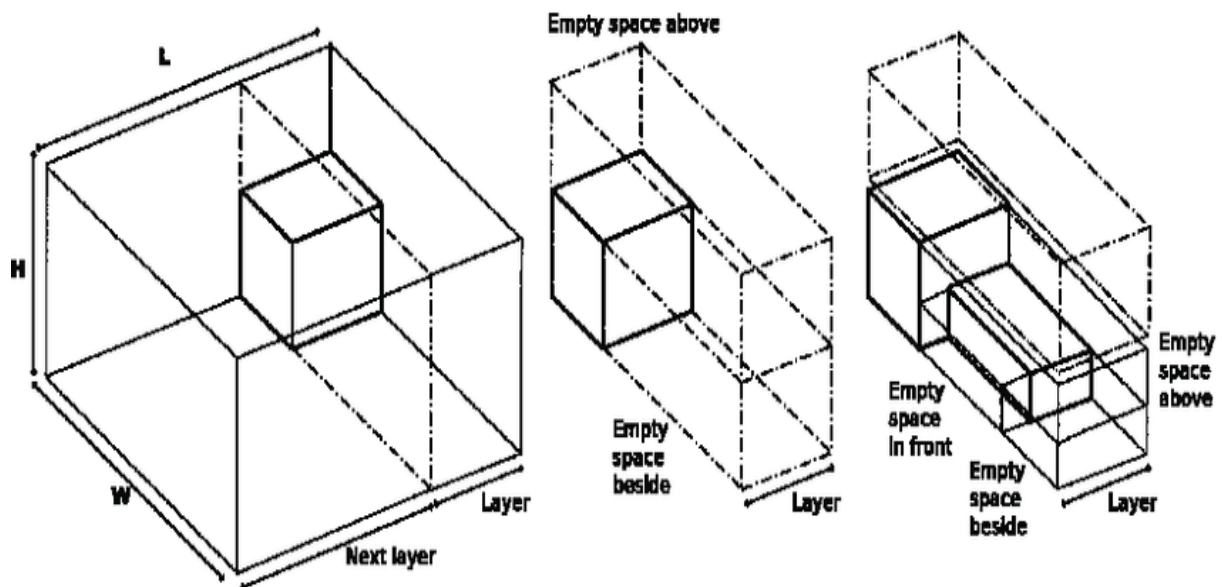


Figure: Création de couche par l'heuristique gourmande

L'exécution de l'algorithme gourmande

La procédure couche de sélection utilise une heuristique de recherche d'arbre à largeur limitée pour déterminer la couche la plus prometteuse, où la largeur est déférente en fonction du niveau de profondeur dans la recherche d'arbre.

L'avantage est que le nombre de nœuds expansés est polynomial et n'est pas de façon exponentielle en ce qui concerne la taille du problème. Après avoir emballé le nombre spécifié de boîtes sur la surface en fonction de la disposition des couches, la surface est divisée en trois sous-surfaces au maximum par les surfaces procédurales divisées de la section V.1.1.1.

Ensuite, la surface d'origine est supprimée de la liste des surfaces disponibles et les sous-surfaces nouvellement générées sont insérées dans la liste. Ensuite, l'algorithme sélectionne la nouvelle surface utilisable la plus basse et répète les procédures ci-dessus jusqu'à ce qu'aucune surface ne soit disponible ou que toutes les boîtes n'aient été emballées dans le conteneur.

Le pseudocode de l'algorithme gourmande (greedy) est donné par la procédure heuristique suivante :

Procédure gourmande (greedy) heuristique () :

Données :

Liste de surface : = surface initiale de $L \times W$ a la hauteur 0

Liste de type de boîte : = tous les types de boîtes

Quand (il existe des surfaces utilisables) **et** (toutes les boîtes ne sont pas emballées)**faire**

Sélectionner la surface utilisable la plus basse en tant que l'ensemble de surfaces actuel

Définir la profondeur: = 0

Définir la meilleure couche: = sélectionner la couche (liste des surfaces, liste des types de boites, profondeur)

Emballer la meilleure couche sur la surface courante

Réduire le nombre des types de boites emballé par le montant emballé

Définir la liste des nouvelles surfaces: = division de surfaces (liste des surface, meilleure couche, liste des types de boites)

Supprimer la surface courante dans la liste des surfaces

Insérer une surface quelconque en liste de nouvelles surfaces

Fin quand

➤ **V.1.1.1 Procédure de division de la surface de chargement en sous-surfaces**

Étant donné une couche de boîtes du même type disposées par l'heuristique gourmande, la couche est toujours placée dans le coin inférieur gauche de la surface de chargement. Comme illustré sur la figure 01 (au-dessous), jusqu'à trois sous-surfaces doivent être créées à partir de la surface de chargement d'origine par la procédure diviser les surfaces, y compris la surface supérieure, qui est au-dessus de la couche qui vient d'être emballée, et les espaces possibles qui pourraient être laissés sur les côtés.

Si $l = L$ ou $w = W$, la surface d'origine est simplement divisée en une ou deux sous-surfaces, la surface supérieure et une surface latérale possible.

Dans le cas contraire, il existe deux variantes de division possibles, à savoir, pour diviser en surface supérieure, la surface (B, C, E, F) et la surface (F, G, H, I), ou à diviser en surface supérieure, la surface (B, C, D, I) et la surface (D, E, G, H). Les divisions sont définies selon les critères suivants:

- Le critère principal est de minimiser la surface totale inutilisable de la variante de division. Si aucune des boîtes restantes ne peut être emballée sur une sous-surface, la zone de la sous-surface est inutilisable.
- Le critère secondaire est d'éviter la création de longues bandes étroites parce que les zones étroites pourraient être difficiles à remplir par la suite.

Plus précisément :

Si : $L - l \geq W - w$, la surface de chargement est divisée à la surface (B, C, E, F) et la surface (F, G, H, I).

Sinon, $L - l \leq W - w$, il est divisé à la surface (B, C, D, I) et la surface (D, E, G, H).

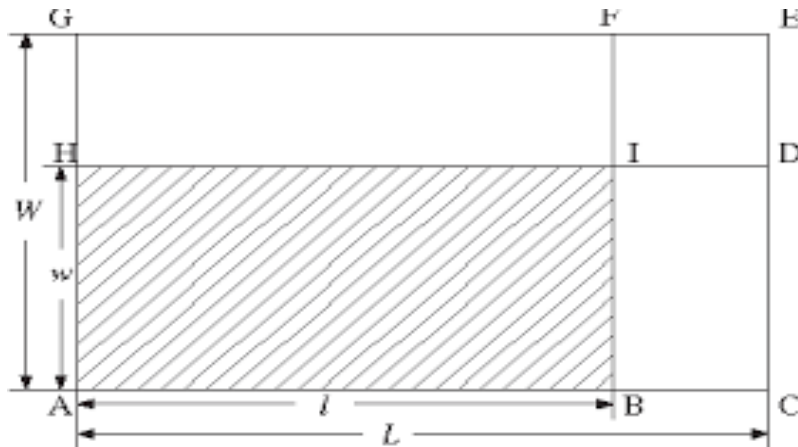


Figure 01 : Division de chargement du surface

➤ V.1.1.2 L'analyse/priorisation

Dans l'étape « d'analyse / priorisation », les types de boîtes considérés comme «maladroits » se voient attribuer plus de reproches, ce qui fait que leurs facteurs de priorité augmentent davantage. Les facteurs de priorité seront ensuite utilisés pour guider l'heuristique gourmande (greedy) dans l'étape de construction.

Il existe plusieurs façons possibles d'analyser la solution et de mettre à jour les facteurs de priorité. Par exemple, une manière possible consiste à augmenter le facteur de priorité de chaque type de boîte par le montant proportionnel au nombre de boîtes qui ne sont pas chargées.

Étant donné que la partie de l'envoi qui n'est pas chargée est relativement petite pour des problèmes de chargement de conteneur unique, la partie non chargée de l'envoi est considérée comme chargée dans un deuxième conteneur supposé et le facteur de priorité de chaque type de boîte est augmenté par un montant égal à $(1 - \text{l'utilisation moyenne du volume du type de boîte})$.

L'utilisation de volume pour chaque boîte simple est définie comme étant l'utilisation de volume du conteneur dans lequel elle est emballée, tandis que l'utilisation de volume d'un conteneur est le rapport de la somme du volume des boîtes emballées sur le volume total du conteneur.

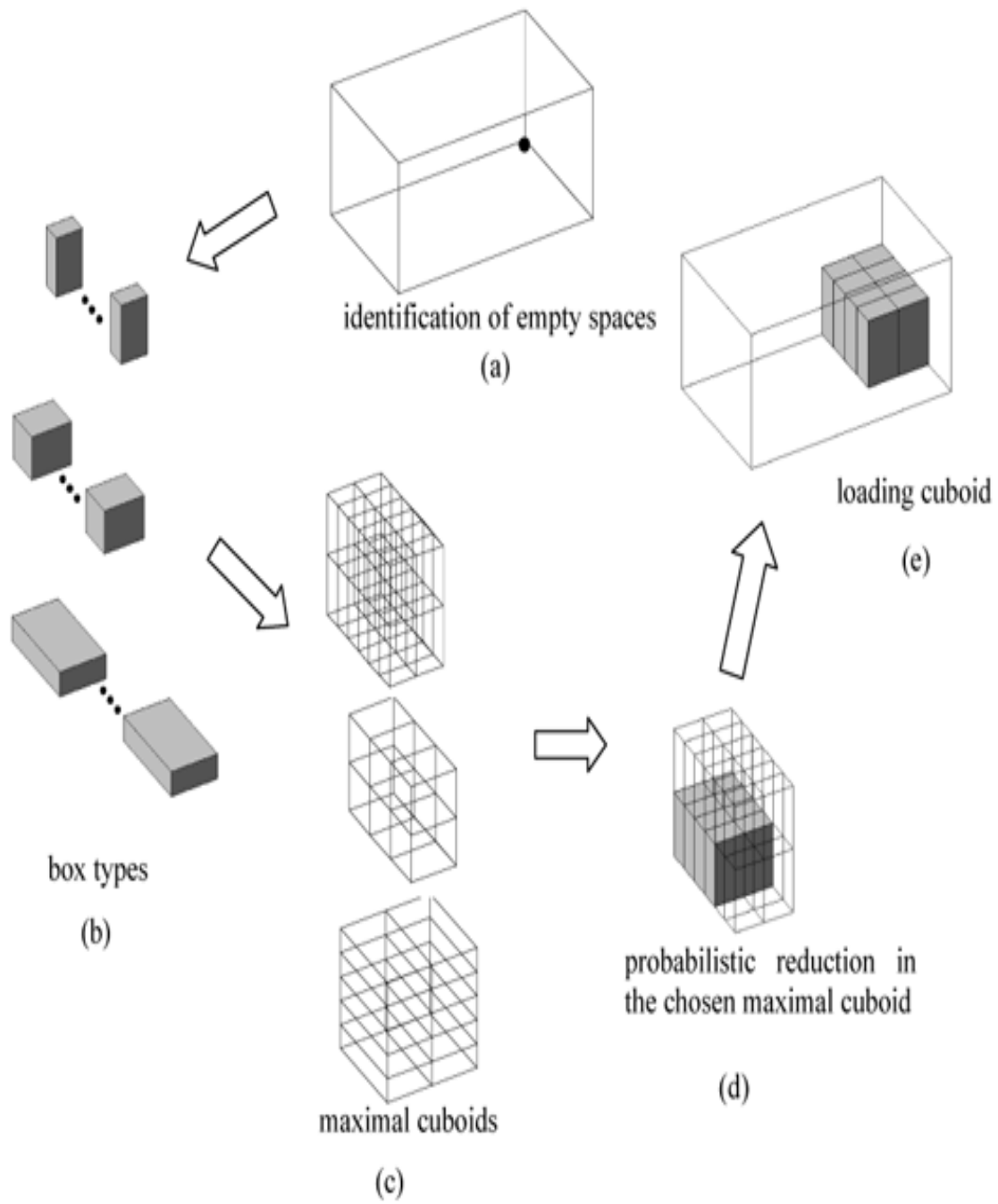


Figure 7 – Schematic representation of the heuristic.

V.1.2- La méthode des matrices

- **Boîtes de même dimensions**

L'objectif de la méthode des matrices c'est de charger un nombre maximum de cartons dans un conteneur et encore il permet un positionnement optimal des cartons pour occuper le volume maximum de conteneur.

L'algorithme des matrices construit le plan de chargement d'un conteneur unique de bas en haut. Cette méthode suit presque le même principe que l'algorithme de gourmande (greedy).

Au stade initial, la liste des volumes disponibles ne contient que le volume initial de taille "L x l" avec sa position initiale à hauteur 0.

- À chaque étape, l'algorithme sélectionne le volume utilisable le plus bas, puis détermine le type de boîte à emballer sur le volume.
- Calcule le nombre de boîte du même type avec la valeur d'évaluation la plus élevée.
 - Soit un conteneur de dimension : (L : longueur, l: largeur, h: Hauteur).
 - On a "m" cartons pour les charger et qui sont de même dimensions.
 - La dimension du carton i est (Li*li*hi).
 - Chaque volume est un nombre positif.
 - Il existe 6 positions différentes pour entreposer les cartons.

Algorithme des matrices pour résoudre ce problème

Données :

Lignes : Li, li, hi (dimension du carton i i=1..m)

Colonnes : L, l, h (dimension de conteneur)

Matrices pour remplir le conteneur :

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$


1)-Saisir les dimensions

2)- Remplissage de la matrice pour déterminer le nombre de carton a charger par largeur, longueur et hauteur

3)- pour chaque combinaison (il existe 6 positions)

-Calcule le nombre de cartons à charger

-Evaluation de l'espace restante dans le conteneur

-Si on trouve un espace pouvant contenir des cartons alors 

On répète l'algorithme avec le nouveau espace comme dimension d'un conteneur fictif.

Application 1

On désire charger des cartons de même dimension dans un seul conteneur. On utilisera la méthode des matrices décrites précédemment.

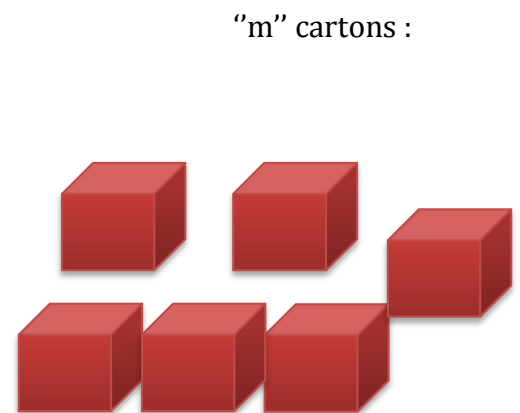
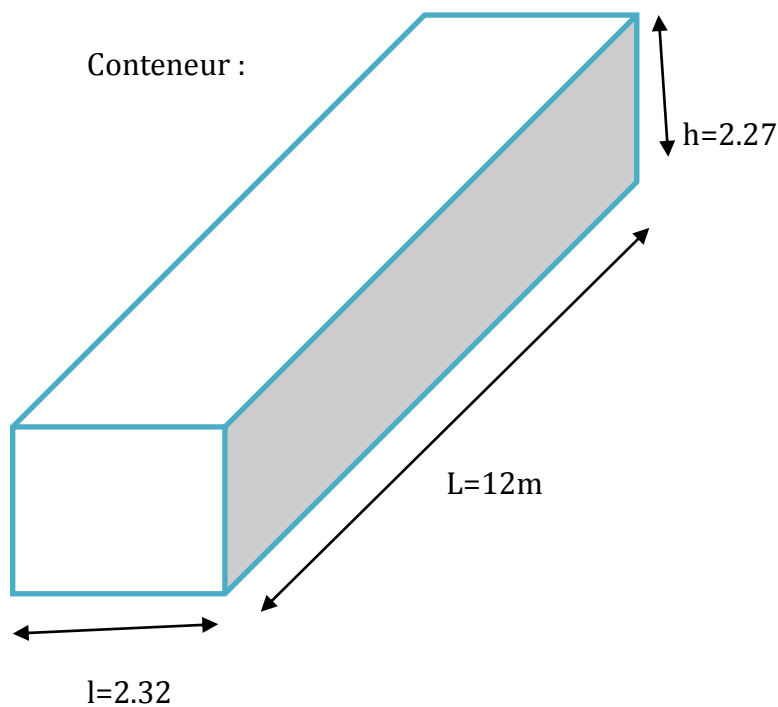
On dispose d'un conteneur et de m boîtes.

- Données

Dimensions de conteneur : $L=12\text{m}$, $l=2.32\text{m}$, $h= 2.27\text{m}$

Dimensions des cartons $Li=0.6\text{m}$, $li=0.6\text{m}$, $hi=0.4\text{m}$

Il existe 6 positions différentes pour entreposer les cartons



2)- Remplissage de la matrice

La matrice nous renseignera sur le nombre de cartons qu'on peut le mettre dans le conteneur, en fonction de la longueur de conteneur, de la largeur ou la hauteur. En appliquant l'opération suivante : $a_{ij} = \text{une des dimension } i \text{ du conteneur} / \text{la dimension } j \text{ d'une boîte}$. (On prend la partie entière du résultat obtenu de cette opération) .Par exemple : $a_{11} = 12/0.6=20$

12 représente la longueur de conteneur et 0.6 longueur du carton

20 sera le nombre de cartons qu'on pourra mettre dans la longueur de conteneur

Avec cette méthode on obtient la matrice suivante de 3 dimensions :

$$\begin{bmatrix} 20 & 3 & 3 \\ 20 & 3 & 3 \\ 30 & 5 & 5 \end{bmatrix}$$

Cette matrice on l'a illustrée avec un tableau :

		Les dimensions de conteneur (m)		
		Longueur	Largeur	Hauteur
		12	2.32	2.27
Les dimensions Des cartons	Longueur	0.6	20	3
	Largeur	0.6	20	3
	Hauteur	0.4	30	5

La 1ère ligne contient les **dimensions du conteneur** et la 1ère colonne celles des cartons.

3)- Calcul du nombre de cartons à charger par combinaisons

On détermine dans cette étape le nombre maximum de cartons pour chacune des 6 positions.

Les six positions des cartons par rapport au conteneur (L, l, h) sont :

a)-Li*li*hi

b)-Li*hi*li

c)-li*LI*hi

d)-li*hi*Li

e)-hi*Li*li

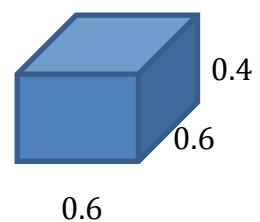
f)-hi*li*LI

-Calcul :

a)-Li*li*hi 0.6*0.6*0.4

(L/Li =12/0.6=20 , l/li=2.32/0.6=3, h/hi=2.27/0.4=5)

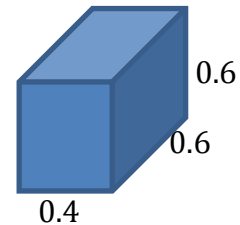
a)= 20*3*5=300 cartons



$$\mathbf{b)-Li*hi*li \quad 0.6*0.4*0.6}$$

($L/Li = 12/0.6 = 20$, $l/li = 2.32/0.4 = 5$, $h/li = 2.27/0.6 = 3$)

$$b) = 20 * 5 * 3 = 300 \text{ cartons}$$



Avec la même procédure on a calculé les autres positions :

$$\mathbf{c)=20*3*5=300 \text{ cartons}}$$

$$\mathbf{d)=20*5*3=300 \text{ cartons}}$$

$$\mathbf{e)=30*3*3=270 \text{ cartons}}$$

$$\mathbf{f)=30*3*3=270 \text{ cartons}}$$

A l'aide de résultats au-dessus. On peut dire jusqu'à maintenant que Le nombre maximum de cartons à charger est donc 300 on peut opter pour le a),b),c)ou d) comme un plan de chargement. La longueur du carton doit correspondre à la longueur du conteneur, la largeur à la hauteur et la hauteur des cartons à la largeur du conteneur.

4)- Evaluation de l'espace restant et comment placer les cartons qu'on va ajouter

Il est parfois possible d'aller au-delà des résultats de la matrice. Il est possible de refaire et parfaire le plan de chargement afin de placer encore plus de cartons. Dans le langage usuel on parle de « correction de la matrice ».

Après l'évaluation de l'espace restant par l'algorithme on trouve qu'on peut ajouter des cartons au conteneur et les nombre de cartons à ajoutés est calculé par cette procédure :

- **Si on prend la position a) (idem pour c)**

Alors : $L = 12\text{m}$ et $20 * 0.6 = 12\text{m}$; $12 - 12 = 0$

$l = 2.32$ et $3 * 0.6 = 1.18\text{m}$; $2.32 - 1.18 = 0.52\text{m}$

$h = 2.27$ et $5 * 0.4 = 2$; $2.27 - 2 = 0.27\text{m}$

Dans ce cas il est possible d'ajouter des cartons au conteneur mais à condition que les cartons qu'on va ajouter on les mit au position b-Li*hi*li ou d -li*hi*Li.

(Les cartons seront disposés de façon à faire correspondre leur hauteur à la largeur de conteneur). On refait une autre matrice à deux dimensions pour voir quelle position (suivant la longueur ou la hauteur) serait plus optimal.

Matrice à 2 dimensions

	12	2.27
0.6	20	3
0.6	20	3

Calculons le nombre des cartons à ajouter :

Dans b) La longueur de carton suit celle de conteneur donc nombre de cartons : 20

La largeur de carton suit la hauteur de conteneur donc le nombre de cartons est :3

Le nombre de cartons que l'on peut ajouter dans a)et c) est : **$20*3=60$** .

Et le total est : $300+60=360$ cartons.

- **Prend la position b) : (idem pour d)**

Alors : $L = 12\text{m}$ et $20*0.6=12\text{m}$; $12-12=0$

$l=2.32\text{m}$ et $5*0.4=2$; $2.32-2=0.32\text{m}$

$h=2.27\text{m}$ et $3*0.6=1.8$; $2.27-1.8=0.47\text{m}$

Alors dans ce cas il est possible d'ajouter des cartons au conteneur mais à condition que les cartons qu'on va l'ajoutés on les mit au position a) et c)

(Les cartons seront disposés de façon à faire correspondre leur hauteur à celle de conteneur). On refait une autre matrice à deux dimensions pour voir quelle position (suivant la longueur ou la largeur) serait plus optimal.

Matrice à 2 dimensions

	12	2.32
0.6	20	3
0.6	20	3

Le nombre de cartons que l'on peut ajouter dans b)et d) :est **$20*3=60$** .

Et le total est : $300+60=360$ cartons.

- **Prend la position e) (idem pour f)**

Alors : $L=12\text{m}$ et $30 \times 0.4=12\text{m}$; $12-12=0$

$l=2.32\text{m}$ et $0.6 \times 3=1.8$; $2.32-1.8=0.52\text{m}$

$h=2.27$ et $0.6 \times 3=1.8$; $2.27-1.8=0.47\text{m}$

Alors dans ce cas il est possible d'ajouter des cartons au conteneur par deux étapes

Etape01

D'ajouter des cartons au conteneur mais à condition que les cartons qu'on va l'ajoutés on les mit au position b)- $Li \times hi \times li$ ou d)- $li \times hi \times Li$.

(Les cartons seront disposés de façon à faire correspondre leur hauteur à la largeur de conteneur). On refait une autre matrice à deux dimensions pour voir quelle position (suivant la longueur ou la hauteur) serait plus optimal.

Matrice à 2 dimensions

	12	2.27
0.6	20	3
0.6	20	3

Le nombre de cartons que l'on peut ajouter dans cette étape est : $20 \times 3=60$

Est le total sera $270+60=330$ cartons

Etape02

Alors dans ce cas il est possible d'ajouter des cartons au conteneur mais à condition que les cartons qu'on va l'ajoutés on les mit au position a) et c)

(Les cartons seront disposés de façon à faire correspondre leur hauteur à celle de conteneur. On refait une autre matrice à deux dimensions pour voir quelle position (suivant la longueur ou la largeur) serait plus optimal.

Matrice à 2 dimensions

	12	2.32
0.6	20	3
0.6	20	3

Le nombre de cartons que l'on peut ajouter dans cette étape est : $20 \times 3 = 60$.

Est le total sera $330 + 60 = 390$ cartons

- On conclut que le nombre maximum de cartons à charger dans le conteneur est donc **390 cartons**. Alors on peut opter pour le e) et f) comme un plan de chargement,

Performance de la méthode

Pour évaluer la performance de la méthode, on compare les résultats obtenus par l'algorithme (S-algor) sur une batterie de données prise d'une manière aléatoire à la solution théorique (S-theo) ; et cela en calculant le ratio : $\mu = \text{S-algor} / \text{S-theo}$ à chaque fois et voir la tendance de μ . Plus il est proche de **1** plus l'algorithme est performant.

Illustrons cette méthode sur notre application.

- La solution théorique

Le volume de chaque carton : $0.6 \times 0.6 \times 0.4 = 0.144 \text{ m}^3$

Le volume de conteneur : $12 \times 2.32 \times 2.27 = 63.19 \text{ m}^3$

Le nombre théorique de cartons qu'on peut mettre dans le conteneur est :

$63.19 / 0.144 = 438 \text{ cartons}$ on a alors **S-theo = 438**

- La solution obtenue par l'algorithme

S-algo = 390 cartons

Calcul de l'optimalité de la solution : $\mu = 390 / 438 = 0.89$

Ce qui permet de conclure que le résultat obtenu par la méthode est assez bon.

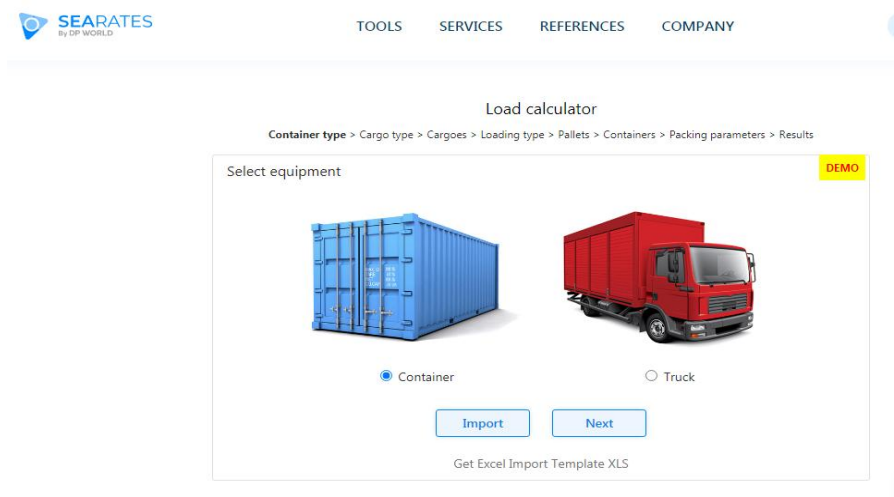
Application 02

Chargement d'un conteneur en utilisant un logiciel « container load calculator »

Présentation du logiciel

Container Load Calculator LCL simule le placement de marchandises à l'intérieur d'un conteneur et détermine leur positionnement optimal à l'intérieur. C'est assez simple: spécifiez le type de cargaison et saisissez ses paramètres (largeur, hauteur, longueur, poids et nombre d'unités). Vous pouvez emballer la cargaison sur des palettes ou directement dans un conteneur de votre choix. L'application affichera la manière la plus optimale de positionner la marchandise. Vous pouvez ensuite personnaliser les résultats comme bon vous semble.

Il est souvent utilisé en tandem avec Cargo Consolidation Service pour déterminer la quantité d'espace que vos marchandises prendront dans un envoi LCL. Les entreprises le combinent avec Trade Platform pour aider à estimer les devis de fret pour l'expédition de marchandises à l'international.



Personnaliser la position, Load Calculator rend une image 3D avec le positionnement optimal des marchandises à l'intérieur d'un conteneur ou d'un camion. Il affiche la meilleure façon de charger des produits de tailles mixtes dans des véhicules et des conteneurs. L'outil est livré avec une poignée de modèles de palettes avec des dimensions, un poids et d'autres paramètres standards.

Vous pouvez personnaliser la position dans laquelle vos caisses, cartons, boîtes ou autres types de marchandises sont chargés. Cette fonction est particulièrement

importante pour le chargement de marchandises dangereuses, fragiles et surdimensionnées.

Load Calculator est disponible dans les langues suivantes : Anglais, Arabe, Allemand, Espagnol, Russe et Chinois.

Résolution de problème de chargement de conteneur avec des boites de dimensions différentes à l'aide du Logiciel LCL

- Un conteneur de dimensions (L = 12m l = 2,32m et h = 2,27m).
- Nombre de cartons m=480 cartons
- La contrainte de poids que ne dépasse pas 28000 kg.
- Il existe **six positions différentes** pour entreposer ces cartons.
- Les cartons sont devisés aux trois types :

Type01 : cargo1 = de dimension (0.2m* 0.4m *0.355m), le nombre de cartons 300

Type02 : cargo2= de dimension (0.6m*0.4m*0.2m), le nombre de cartons 80

Type03 : cargo3= de dimension (2m*0.6m*0.445m), le nombre de cartons 100

La même question à poser c'est de savoir comment les positionner dans le conteneur pour en charger le maximum possible ?

Et est-ce-que on peut charger tous les cartons dans ce conteneur et en tenant la considération la contrainte du poids ?


Alors à l'aide de logiciel « container load calculator » on a réussi à trouver la solution Maintenant on mit la solution avec ces étapes qu'on la fait sur le logiciel


Etape 01 : choisir le conteneur pour le charger

Load calculator

Container type > Cargo type > Cargoes > Loading type > Pallets > Containers > Packing parameters > Results

Select equipment DEMO


 Container


 Truck

ImportNext

[Get Excel Import Template XLS](#)


Etape 02 : choisir la forme des cargaisons à charger

Select cargo type


[Container type](#) > **Cargo type** > Cargoes > Loading type > Pallets > Containers > Packing parameters > Results

Select cargo type DEMO


Select packing type.




Boxes



BigBags



Barrels



Sacks


Back
Next

Etape03 : completer le tableau par les dimensions des cargaisons et le poids de chacune et le nombre de boites de chaque type

Cargo parameters DEMO

Enter cargo data for stuffing.


You can enter not more than 30 cargo types.



Boxes
Add cargo

Cargo dimensions have to be entered in millimeters (mm) and cargo weight in kilogram (kg)!

You are viewing the Demo calculation mode. To change package type, sizes or add more cargoes to optimize your stuffing - please go to the [subscription](#)

	Name	Length (mm)	Width (mm)	Height (mm)	Weight (kg)	Q-ty	Color
×	Cargo1	200	400	355	30	300	Orange ▼
×	Cargo2	600	400	200	50	80	Green ▼
×	Cargo3	2000	600	440	60	100	Tomato ▼


Back
Next

Etape04 : choisir la façon de charger le conteneur


Type of loading

[Container type](#) > [Cargo type](#) > [Cargoes](#) > **Loading type** > [Pallets](#) > [Containers](#) > [Packing type](#) > [Results](#)

Select equipment for cargo loading DEMO



Cargo stuffing directly into container.






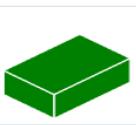

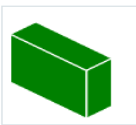



Pre-shipment cargo palletizing.

Back Next

Etape05 : on sélectionne les orientations qu'on peut effectuer à chaque cargaison

Spacing settings of cargo in container DEMO

Select position(s) of load acceptable for the transportation of that cargo type. Putting limits on positions of load could adversely affect effective stuffing.

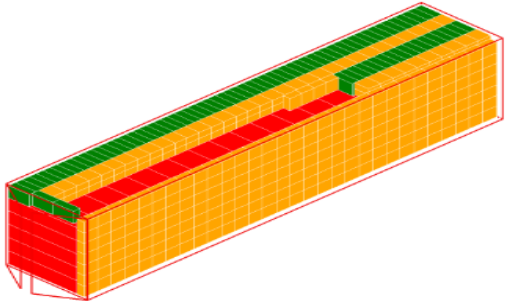

<p>Cargo: Cargo1 Q-ty: 300 units</p> <p>Weight: 30 kg. Length: 200 mm. Width: 400 mm. Height: 355 mm.</p>	<p>H e i g h t</p>  <p><input checked="" type="checkbox"/></p>	<p>L e n g t h</p>  <p><input checked="" type="checkbox"/></p>	<p>W i d t h</p>  <p><input checked="" type="checkbox"/></p>
<p>Cargo: Cargo2 Q-ty: 80 units</p> <p>Weight: 50 kg. Length: 600 mm. Width: 400 mm. Height: 200 mm.</p>	<p>H e i g h t</p>  <p><input checked="" type="checkbox"/></p>	<p>L e n g t h</p>  <p><input checked="" type="checkbox"/></p>	<p>W i d t h</p>  <p><input checked="" type="checkbox"/></p>
<p>Cargo: Cargo3 Q-ty: 100 units</p> <p>Weight: 60 kg. Length: 2000 mm. Width: 600 mm. Height: 440 mm.</p>	<p>H e i g h t</p>  <p><input checked="" type="checkbox"/></p>	<p>L e n g t h</p>  <p><input checked="" type="checkbox"/></p>	<p>W i d t h</p>  <p><input checked="" type="checkbox"/></p>

Back Next

Etape06 : la solution

Stuffing result

[Container type](#) > [Cargo types](#) > [Cargoes](#) > [Loading types](#) > [Pallets](#) > [Containers](#) > [Packing parameters](#) > **Results**

Results		DEMO
Containers	Cargo packing	
All containers: 40' hq: 1 units	Total: 480 packages. Packed: 480 packages. (100%)	
	Container N°1 (40' hq 1 units) Packed: 480 packages: (100%). Including: <ul style="list-style-type: none">Cargo1 - 300 packages (100%)Cargo2 - 80 packages (100%)Cargo3 - 100 packages (100%)	
	Cargo volume: 65.16 m ³ (85% of volume) Cargo weight: 19000 kg. (66% of max payload) Show packing by blocks Show packing step by step 	

[Back](#) [Export .dat](#) [Export .xls](#)

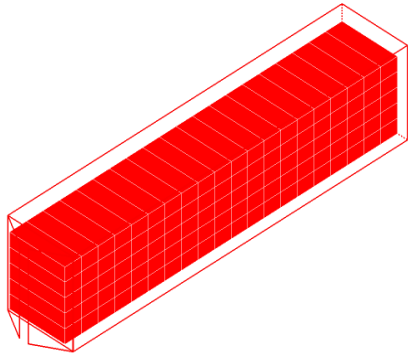
De cela on conclut que toutes les boites sont chargées de façon optimal dans le conteneur et leurs poids ne dépasse pas la contrainte de départ ou :

Poids de cargaison 19000 kg < poids limites de conteneur 28000kg

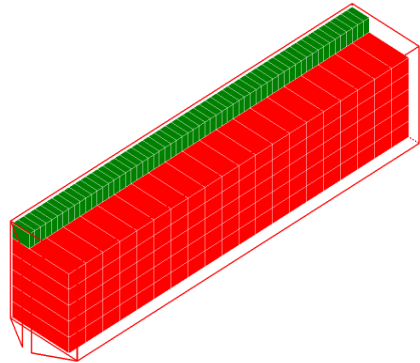
Encore on a obtenu cette solution dans un intervalle de temps très court donc ce logiciel est performant.

On peut illustrer au-dessous comment le conteneur a été rempli :

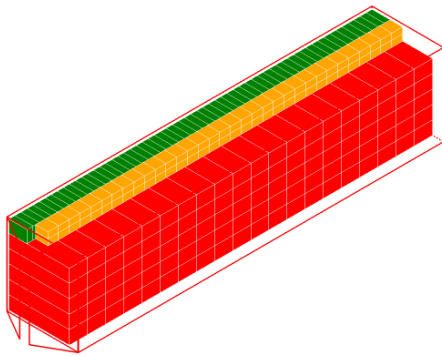
Stuffing result
Block №1



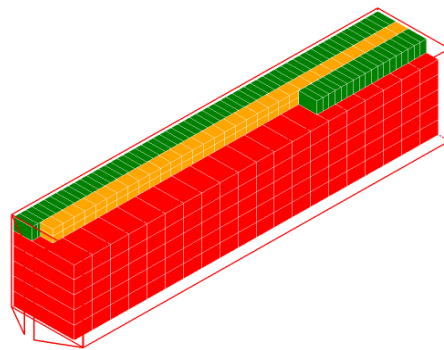
Block №2



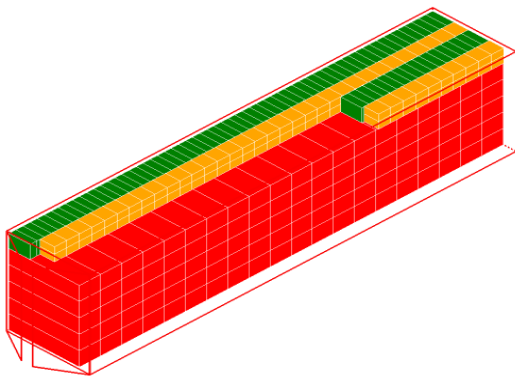
Block №3



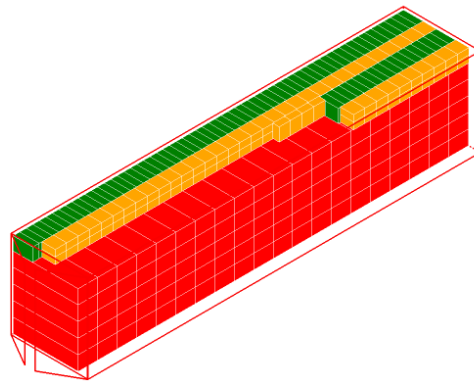
Block №4



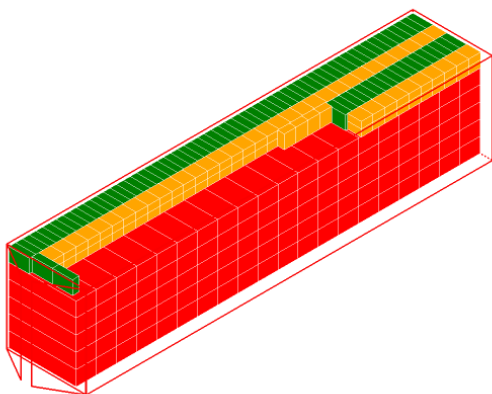
Block №5



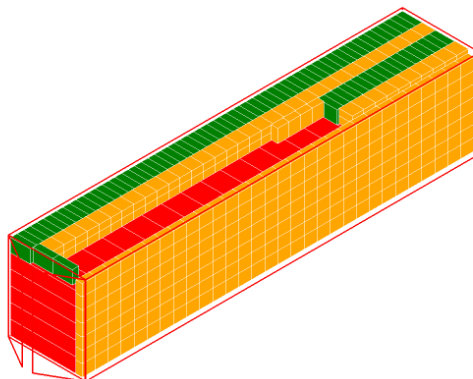
Block №6



Block №7



Block №8



Application 03

Algorithme pour le problème de chargement de plusieurs conteneurs

Afin de nous familiariser avec la recherche, on s'est intéressé à l'étude d'un article : « un examen de l'état de l'art » édité dans la revue « journal de recherche opérationnelle ». Écrit par les auteurs Bischoff et Ratcliff.

La construction heuristique gourmande pour les problèmes de chargement de conteneurs uniques est modifiée et étendue pour résoudre le problème de chargement de conteneurs multiples en utilisant l'approche séquentielle.

C'est-à-dire que les conteneurs sont remplis les uns après les autres. L'approche tente de remplir un seul conteneur avec la valeur d'évaluation la plus élevée par l'heuristique gourmande pour les problèmes de chargement de conteneur unique. L'heuristique de construction se poursuit avec la réduction de la sous-réservation et un nouveau conteneur vide jusqu'à ce que la totalité de boîtes soit chargée.

Les raisons de l'utilisation de l'approche séquentielle, au lieu de l'approche simultanée, incluent l'utilisation de la priorisation, la supériorité de l'approche séquentielle sur l'approche simultanée pour les algorithmes de construction de couches et la facilité de mise en œuvre.

L'existence de types de boîtes « problématiques » est également observée dans des multiples problèmes de chargement de conteneurs [15]. L'existence de points chauds signifie que l'idée de priorisation pour le problème de chargement d'un conteneur unique pourrait être utile pour les problèmes de chargement de conteneurs multiples. Dans l'approche séquentielle les emballages les uns après les autres ont un inconvénient possible est que les boîtes utilisées lors de l'emballage des premiers conteneurs pourraient être mieux utilisées pour emballer les conteneurs ultérieurs.

D'un autre côté, les boîtes qui doivent être emballées dans des conteneurs antérieurs sont laissées dans des conteneurs plus longs, ce qui fait que les conteneurs ultérieurs ont une mauvaise utilisation du volume.

Ainsi, l'approche pourrait réduire le nombre de conteneurs utilisés. L'excellente intégration de la priorisation avec l'approche séquentielle est également vérifiée par de bons résultats expérimentaux dans la section suivante.

La solution trouvée par l'heuristique gourmande (greedy) est analysée et les facteurs de priorité sont mis à jour par le même analyseur et prioriseur utilisé dans le problème de chargement de conteneur unique, puis l'heuristique gourmande est appliquée à nouveau avec de nouveaux facteurs de priorité et les procédures ci-dessus sont répétées. Ce cycle continue jusqu'à ce que le nombre maximal d'itérations soit atteint ou que le délai soit dépassé.

Résultats expérimentaux

Les algorithmes pour les problèmes de chargement de conteneurs, y compris les cas hétérogènes simples et les cas hétérogènes multiples, ont été mis en œuvre en Java. Les expériences ont été réalisées sur un PC Pentium 4 2,40 GHz. En appliquant l'algorithme IC-Algorithm (Iterated Construction Algorithm) dans les sections suivantes :

Résultats expérimentaux pour les problèmes de chargement de conteneurs hétérogènes simples

L'approche pour les hétérogènes simples les problèmes de chargement de conteneurs ont été testés sur des instances de référence standard de 700 instances de Bischoff et Ratcliff. Ces références sont disponibles en ligne dans la bibliothèque-OR. Les 700 cas de test Bischoff/ Ratcliff sont divisés en sept classes de test, à savoir BR1 à BR7, où chaque classe de test se compose de 100 cas de test. L'approche a été comparée aux approches de Bischoff, Janetz et Ratcliff, 1995 (B / J / R).

De Bischoff et Ratcliff, 1995a (B / R).

De Gehring et Bortfeldt, 1997 (G / B).

De Bortfeldt et Gehring, 1998 (B / G).

De Davies et Bischoff, 1999 (D / B).

De Terno Scheithauer, Sommerweiß et Riehme, 2000 (T / S / S / R).

De Bortfeldt et Gehring, 2001 (B / G) et d'Eley, 2002 (E).

<i>Problem</i>	Minimum	Average	Maximum	SD
BR1(3)	79.18	91.60	97.06	2.87
BR2(5)	86.61	91.99	96.26	1.93
BR3(8)	87.84	92.30	94.67	1.19
BR4(10)	89.70	92.36	94.79	1.05
BR5(12)	88.88	91.90	94.33	0.93
BR6(15)	89.18	91.51	93.76	0.86
BR7(20)	88.85	91.01	93.11	0.90

Tableau 01: utilisation de volume (%) pour BR1 jusqu'à BR7 avec : IC-Algorithm

Approche	Utilisation de volume (%)
B/J/R(1995)	81.97
B/R(1995a)	83.45
B/R-comb	84.94
G/B(1997)	88.30
B/G(1998)	91.26
D/B(1999)	84.19
T/S/S/R(2000)	88.51
B/G(2001)	90.06
E(2002)	88.75
IC(2003)	91.81

Tableau 02: Utilisation moyenne du volume pour BR1 à BR7

De plus, le peigne B / R combine la valeur maximale de B / J / R et B / R pour n'importe quel cas de test. Les résultats expérimentaux pour les 700 tests Bischoff / Ratcliff sont résumés dans **le tableau 01**. Les utilisations de volume minimal, moyen et maximal de 100 instances de test pour chaque classe de test sont présentées, ainsi que les écarts-types des instances. Pour toutes les classes de test, l'utilisation moyenne du volume reste supérieure à 91%. Dans **le tableau 02**, l'utilisation du volume moyen global pour les 700 cas de test est donnée et comparée aux utilisations obtenues par d'autres méthodes.

Les utilisations en volume d'autres méthodes sont tirées de la littérature et mises à jour avec les publications les plus récentes. Parmi les 10 approches, cette approche a atteint l'utilisation de volume la plus élevée, environ 0,55% plus élevée que la deuxième. Le temps d'exécution moyen des 700 cas de test est de 706,59 secondes, tandis que les temps minimal et maximal sont respectivement de 0,89 seconde et 4664,30 secondes.

Conclusion

Ils ont développé un nouvel algorithme pour le problème de chargement de conteneur unique et multiple et ils ont développés l'algorithme de chargement de conteneur unique.

L'algorithme utilise la priorité dynamique pour gérer les types de boîtes problématiques. Le type de boîte(box) avec une priorité plus élevée est emballé sur les surfaces inférieures du conteneur. La solution trouvée par la construction heuristique est analysée et les priorités sont mises à jour et utilisées lors de la prochaine exécution de l'heuristique de construction. L'algorithme est modifié et étendu pour répondre à plusieurs problèmes de chargement de conteneurs en utilisant l'approche séquentielle.

Et on a terminé avec des exemples d'applications pour illustrer cet algorithme.

Conclusion générale

Dans notre mémoire on a étudié les problèmes de chargement de conteneurs qui sont des problèmes d'optimisation combinatoire avec de nombreuses applications dans divers domaines.

Le domaine de chargement de conteneurs et 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 chargement de conteneurs, on s'est attardé au placement des boites forme rectangulaire, dans un conteneur, de sorte qu'aucune de ces boites ne se chevauchent et toutes les boites sont placés en parallèle à l'axe de conteneur.

L'objectif est surtout de maximiser le taux d'utilisation de volume de conteneur.

Alors que plusieurs algorithmes et heuristiques se présentent, après plusieurs recherche et compréhension, notre étude nous a conduit à choisir l'heuristique gourmande(greedy) et l'algorithme de la méthode des matrices que nous avons développés et appliqués à des cas pratique, puis on a choisi un logiciel « container load calculator » pour se familiariser avec l'outil informatique pour la résolution de ce problème, ça nous a aussi permet de comparer les résultats obtenus par les deux méthodes de résolution. Afin de nous initier au travail de recherche on a fini notre travail par étudier et analyser un article traitant de problème.

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 difficiles au sens de la complexité.

La programmation de cette méthode est vivement souhaitée et qui peut faire l'objet de sujet de mémoire intéressant dans l'avenir.

Bibliographie

- [1] Thèse de doctorat « Utilisation des Approches d'Optimisation Combinatoire pour La Vérification des Applications Temps Réel ». Réalisé par Abdesslem LAYEB,, Université Mentouri de Constantine 2010.
- [2] Thèse de master « Etude de problème de rangement ouvert avec conflits ». Réalisé par B. R. Tebbal mohamed 2016.
- [3] « Cours des Méthodes de Résolution Exactes Heuristiques et Méta heuristiques », par Ben said Omar Université Mohammed V, Faculté des Sciences de Rabat.
- [4] Thèse « la complexité algorithmique ». Réalisé par Nadia Brauner
- [5] Thèse « Complexité Algorithmique » réalisé par B. Mariou 2012.
- [6] « complexité », fichier Licence Informatique Algorithmique et Programmation, par Roy Peter 2015 .
- [7] "Handbok of combinatorial optimization". Springer, D.Du and P.M Pardalos ,2007
- [8]"Cutting and Packing in Production and Distribution; A Typology and Bibliography, ed. Physica-verlag, A Springer-Verlag", H. Dyckhoff et U. Finke 1992.
- [9] Mémoire magister « résolution numérique dans les problèmes de type bin packing » réalisé par Amara Aida kenza
- [10]"New approaches to the pallet loading problem". Nelißen.
- [11]"Heuristic for the rectangle-packing problem. European Journal of Operational Research" G. Scheithauer and U. Sommerweiß., 108:509–526, 1998.
- [12]. The heuristic for the pallet loading problem. Journal of the Operational Research Society, G. Scheithauer and J. Terno 47:511–522, 1996.
- [13] "Issues in the development of approaches to container loading.Omega International Journal of Management Science ",E. E. Bischoff and M. S. W. Ratcliff,23:377–390, 1995.
- [14]"An efficient approach for the multi-pallet loading problem. European Journal of Operational Research",Terno, G. Scheithauer, U. Sommerweiß, and J. Riehme, 123:372–381, 2000
- [15]"Loading multiple pallets. Journal of the Operational", E.E. Bischoff and M.S.W. Ratcliff Research Society, 46:1322–1336, 1995