Rapport du TP OPTIMISATION Pt.1: Methodes Exactes

BACHI Yasmine (CdE) SAADI Fatma Zohra Khaoula
NOUALI Sarah MOUSSAOUI Meroua
MIHOUBI Lamia Zohra

30 avril 2020

Table des matières

Introduction

Le problème du bin packing, dans lequel un ensemble d'objets de différents poids doit être rangé dans un nombre minimum de boîtes identiques de capacité C est un problème NP-difficile, c'est à dire qu'il n y a aucune chance de trouver une méthode de résolution qui fournit la solution exacte en un temps polynomiale, sauf si l'égalité NP=P est prouvée. Durant le dernier siècle, divers efforts ont été consacrés pour étudier ce problème, dans le but de trouver des algorithmes heuristiques rapides pour fournir de bonnes solutions approximatives. Dans ce projet, nous allons mettre en place un plateforme de résolution du problème du Bin Packing. pour cela , nous implémenterons 4 types de méthodes :

1. *méthodes exactes :* fournissant la solution optimale, mais qui sont très limités par la taille du problème.
2. *heuristiques :* qui sont des méthodes approchées spécifiques au problème.
3. *métaheuristiques :* qui sont des méthodes approchées génériques.
4. *hybridation d'une métaheuristique avec une recherche locale :* qui est notre contribution principale dans la résolution de ce problème.

Nous commencerons par la présentation du problème, sa formulation mathématique, et une étude des méthodes de résolutions existantes dans la littérature.[Etat de l'Art]. Ensuite, nous présenterons la conception détaillée de chaque méthode implémentée, ainsi que les résultats des test de ces méthodes effectués sur des benchmark connus.[Conception & Tests] On distingue 2 types de tests :

1. *les tests empiriques :* dont le but de trouver la meilleure configuration des paramètres de nos méthodes implémentées.
2. *les tests comparatifs :* où on doit comparer les résultats obtenus des méthodes implémentées et sélectionner la meilleure méthode de résolution pour chaque instances. la comparaison se fait en terme de qualité de la solution et du temps d'exécution.

Chapitre 1

Présentation du Problème de Bin Packing (BPP)

1.1 Domaines d'Application :

Le BPP a de nombreuses applications dans le domaine industriel, informatique, etc. Parmi lesquelles on trouve :

- Chargement de conteneurs.
- Placement des données sur plusieurs disques.
- Planification des travaux.
- Emballage de publicités dans des stations de radio / télévision de longueur fixe.
- Stockage d'une grande collection de musique sur des cassettes / CD, etc.

1.2 Formulation Mathématique

Etant donné m boites de capacité C et n articles de volume v_i chacun. Soient :

$$x_{ij} = \begin{cases} 1 & \text{article } j \text{ rangé dans la boîte } i \\ 0 & \text{sinon} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{boîte } i \text{ utilisée} \\ 0 & \text{sinon} \end{cases}$$

La formulation du problème donne ainsi le programme linéaire suivant

$$(PN)\begin{cases} Z(min) = \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_{ij} = 1 \\ \sum_{j=1}^{n} v_j x_{ij} \leq C y_i \\ y_i \in \{0,1\} \\ x_{ij} \in \{0,1\} \end{cases}$$

La première contrainte signifie qu'un article j ne peut être placé qu'en une seule boite La deuxième fait qu'on ne dépasse pas la taille d'une boite lors du rangement

Chapitre 2

Etat de l'Art

Après une étude ciblée des travaux existants sur le problème du Bin Packing, nous avons abouti à une synthèse des méthodes de résolution les plus connues –dans chacune des trois catégories : méthodes exactes, et méthodes approchées : heuristiques et métaheuristiques– que nous allons exposer dans ce qui suit.

2.1 Méthodes Exactes

Les méthodes exactes permettent d'avoir des solutions optimales, cependant le temps de calcul peut être très long pour certaines instances du problème. Il n'existe pas un grand nombre de méthodes exactes pour résoudre le problème du Bin Packing , nous allons présenter dans ce qui suit la méthode MTP (basée sur le Branch and Bound) et une méthode de programmation dynamique DP-flow.

2.1.1 Branche and Bound

Cette méthode a été utilisée pour la première fois dans les années cinquante pour résoudre qui a été modélisé par un programme linéaire en nombres entiers. Afin de rendre le processus de résolution plus rapide, cet algorithme utilise une borne inférieure. Plusieurs techniques ont été proposés pour obtenir cette dernière.

Borne inférieure évidente : Soient C la capacité des boîtes utilisés, A l'ensemble des articles a_i de volumes v_i de l'instance I.

$$BI(I) = \frac{\sum_{i=1}^{n} v_i}{C}$$

Borne de Martello and Toth L2 : Soit α un entier tels que :

$$0 \leq \alpha \leq C/2$$

On definie des classes d'articles suivantes :

$$C_1 = \{a_i, \quad C - \alpha < v_i\}$$
$$C_2 = \{a_i, \quad C/2 < v_i \leq C - \alpha\}$$
$$C_3 = \{a_i, \quad \alpha < v_i \leq C/2\}$$

$BI(I)$ est donnée par la formule suivante :

$$BI(I) = max\{L(\alpha), \quad 0 \leq \alpha \leq C/2\}$$

Avec

$$L(\alpha) = |C_1| + |C_2| + max(0, \left\lceil \frac{\sum_{j \in C_3} v_j - (|C_2| * C - \sum_{j \in C_2} v_j)}{C} \right\rceil)$$

cette borne est calculé en un temps $o(nlogn)$.

Borne inférieure L_3 : Une autre technique a été utilisé dans l'algorithme de résolution MTP pour déterminer une borne inférieure. Soient n_1 le nombre de boîtes obtenus après la première application de la technique de réduction MTP qui consiste à réduire l'instance du problème en rangeant l'article le plus petit, soit Ir^1 l'instance résiduelle de l'instance I après cette première application ie l'ensemble des articles restants après l'opération de réduction

$$L'_1 = n_1 + L_2(Ir^1) \geq L_2(I)$$

On refait ce processus jusqu'à ce que l'instance résiduelle soit vide (i.e. : tous les articles ont été rangés). A l'itération k, on obtiendra :

$$L'_K = \sum_{i=1}^{k} n_i + L_2(Ir^k)$$

La borne inférieure L_3 est obtenue en appliquant la formule suivante , par la suite :

$$L_3 = max\{L'_1, L'_2, ..., L'_{kmax}\}$$

Un des algorithmes proposés pour cette méthode est l'algorithme MTP :

L'algorithme MTP (Martello and Toth Procédure) :

Le meilleur algorithme existant pour trouver la solution optimale du problème Bin Packing est celui proposée par *Martello et Toth (Martello & Toth 1990a ; 1990b)* le principe est le suivant : Les articles sont initialement triés selon des poids décroissants. À chaque nœud de décision, le premier élément libre est attribué, à son tour, aux boîtes existantes qui peuvent le contenir (on parcourt les boîtes par ordre de création) et à une nouvelle boîte. À chaque nœud de l'arbre de recherche

a. Une borne inférieure L_3 de la solution restante est calculée et utilisée pour élaguer le nœud de l'espace de recherche et réduire le problème actuel.
   - Si la borne inférieure du nœud actuel est supérieure à la borne inférieure du problème d'origine (nœud racine), le nœud est supprimé. Sinon (b)

b. Des algorithmes approximatifs FFD, BFD et WFD (qu'on présentera dans la partie Méthodes approximatives) sont appliqués au problème actuel, et chacune des solutions approximatives obtenues est comparée à la borne inférieure L_3.
   - Si le nombre de boîtes utilisées par l'une des solutions approximatives est égal à la borne inférieure du nœud actuel, aucune autre recherche n'est effectuée sous ce nœud.
   - Si le nombre de boîtes utilisées dans une solution approximative est égal à la borne inférieure L3 du problème d'origine (nœud racine), l'algorithme se termine, renvoyant cette solution comme optimale.

De plus, La principale source d'efficacité de l'algorithme de Martello et Toth est une méthode pour réduire la taille des sous-problèmes restants, appelée **critère de dominance**.

2.1.2 Programmation Dynamique