

Optimisation & Operational Research

Bin Packing, approx algorithms and B&B

Arnaud Vandaele

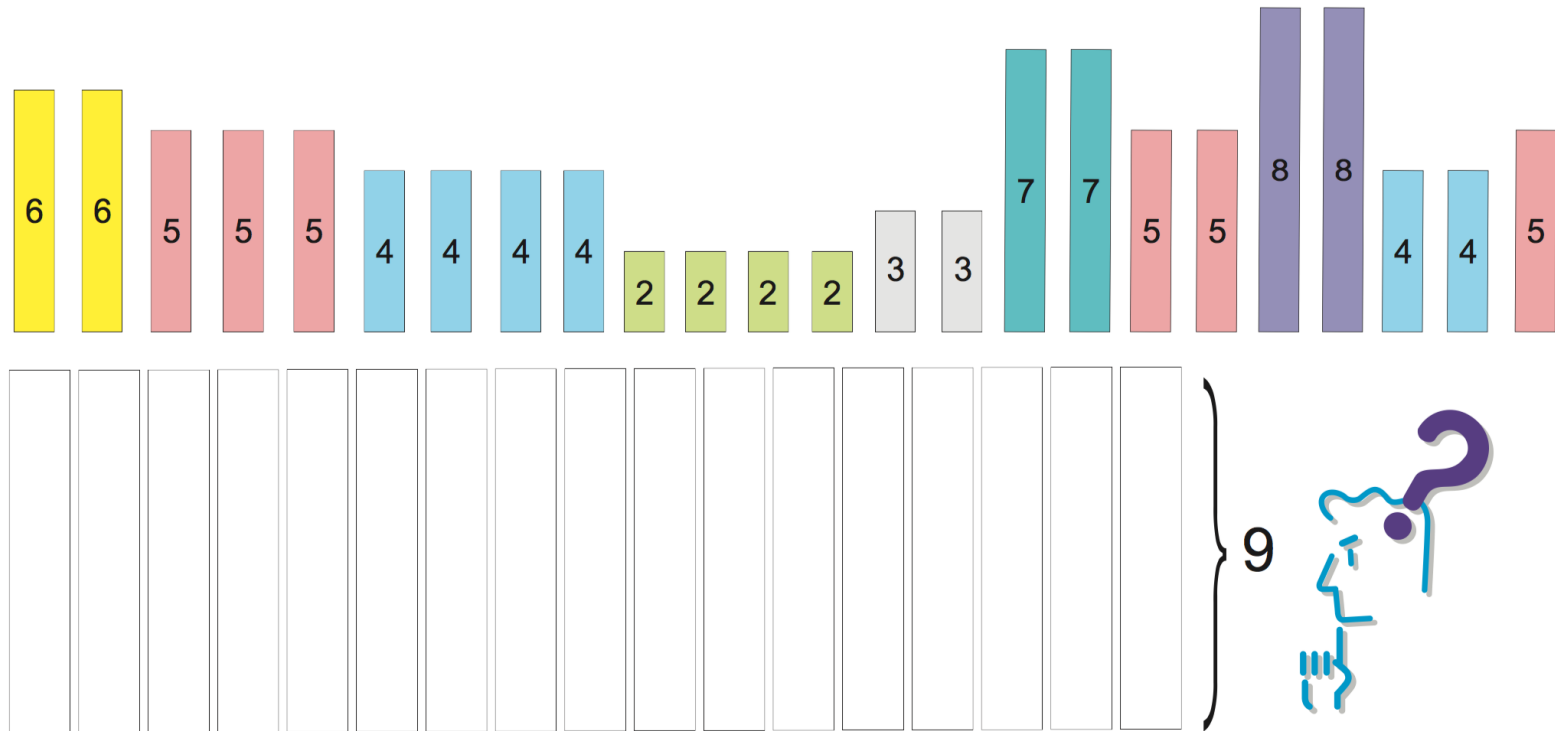
UMONS – FPMs – MathRO

Sommaire

- Bin Packing & Cutting Stock problem : descriptions
- Formulations mathématiques
- Preuve de NP-hardness
- Algorithmes d'approximation
- Branch and Bound

Bin Packing

Vous êtes la personne en charge de l'emballage dans une grande entreprise. Votre travail consiste à emballer habilement des articles de différentes tailles dans une boîte d'une longueur prédéterminée ; votre objectif est d'utiliser le moins de boîtes possible. Comment ces articles doivent-ils être emballés ?

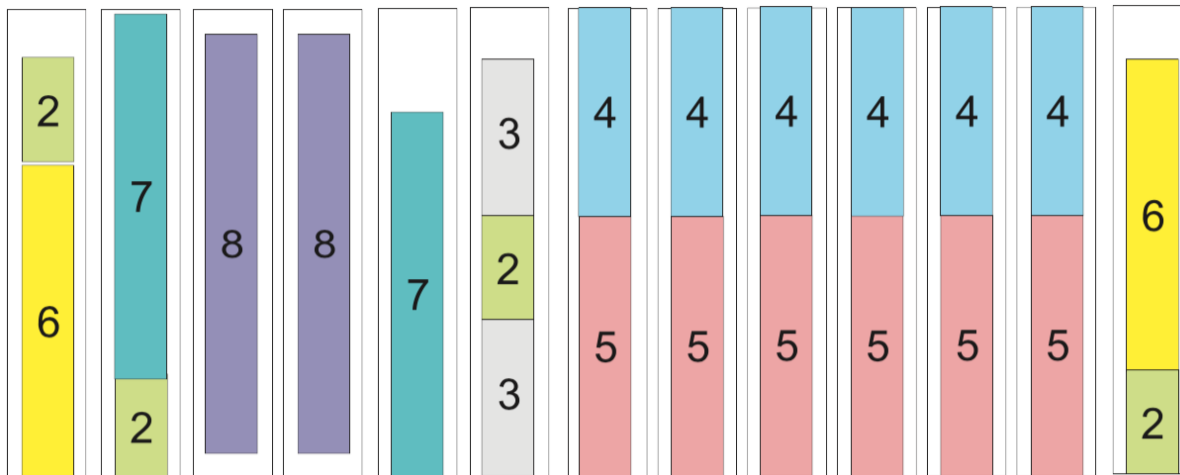


Bin Packing

Définition formelle :

En utilisant des boîtes (bins) de longueur C , comment assembler le mieux possible un ensemble de m objets de taille s_1, \dots, s_m (avec $0 < s_i \leq C$) de façon à utiliser le moins de boîtes possible ?

Pour l'exemple du slide précédent, avec $C=9$, $m=24$ et $s = \{6, 6, 5, 5, 5, 4, 4, 4, 4, 2, 2, 2, 2, 3, 3, 7, 7, 5, 5, 8, 8, 4, 4, 5\}$, la solution optimale utilise 13 boîtes :



Bin Packing - Formulation

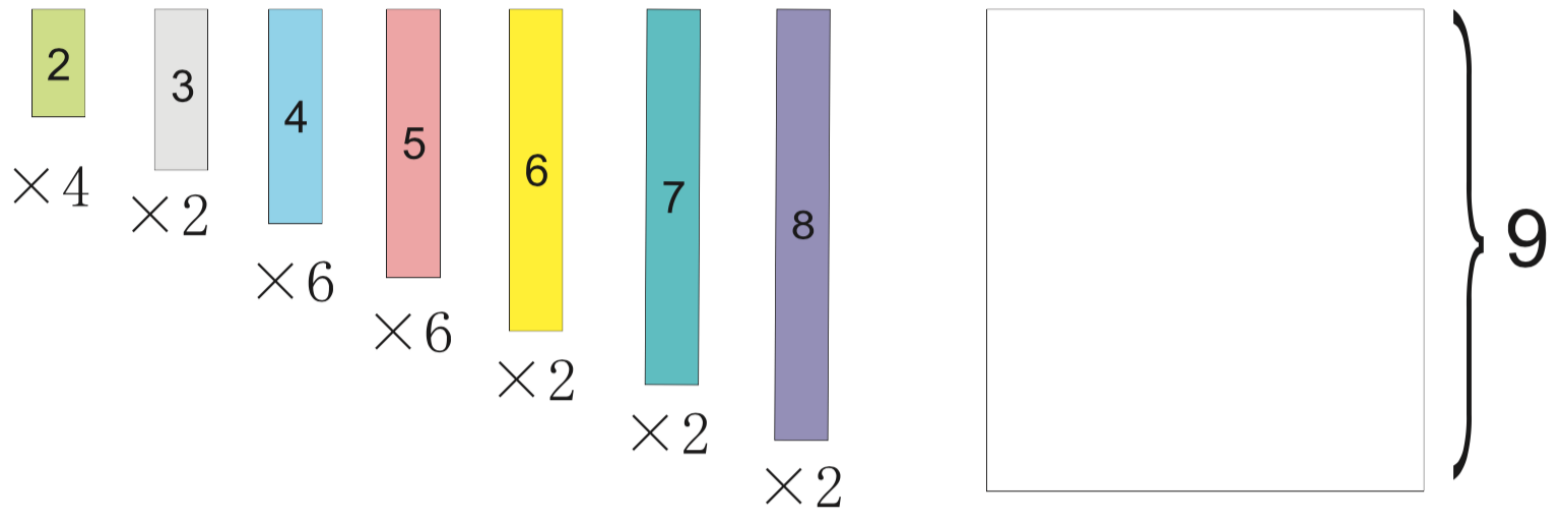
Définition formelle :

En utilisant des boîtes (bins) de longueur C , comment assembler le mieux possible un ensemble de m objets de taille s_1, \dots, s_m (avec $0 < s_i \leq C$) de façon à utiliser le moins de boîtes possible ?

Cutting Stock Problem

Un problème semblable au Bin Packing est le Cutting Stock Problem.

Vous êtes le responsable de la découpe dans une grande entreprise produisant des rouleaux de papier. Votre travail consiste à couper habilement les gros rouleaux produits dans une taille standard en petits rouleaux, avec les tailles demandées par les clients. L'objectif est d'utiliser le moins de gros rouleaux possible ; en d'autres termes, de minimiser la perte. Comment les grands rouleaux doivent-ils être coupés ?



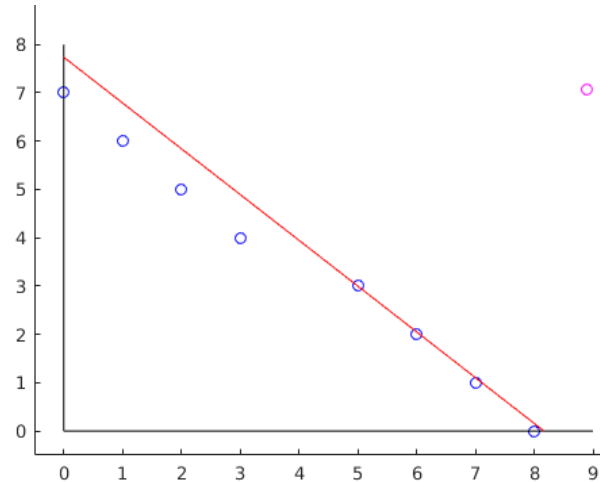
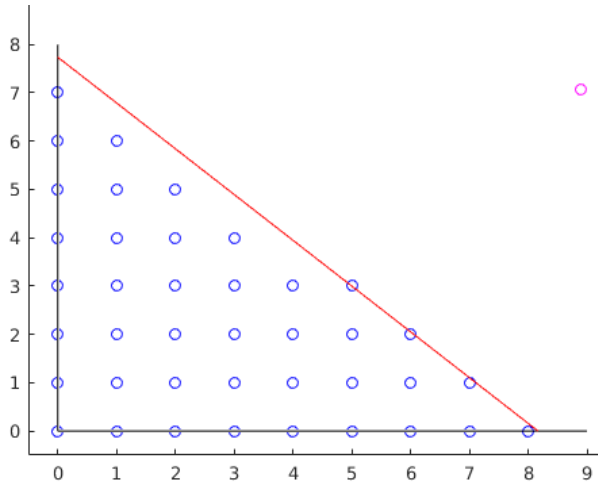
Bien que formulé différemment, ce problème est équivalent au Bin Packing.

Cutting Stock Problem

Définition formelle :

En utilisant des rouleaux de longueur standard C , on souhaite, en les découpant, di pièces de longueur s_i (avec $0 < s_i \leq C$), $i=1, \dots, n$. Comment organiser ces découpes afin de minimiser le nombre de rouleaux standards utilisé ?

Approche : on génère tous les patterns possibles (toutes les combinaisons de longueur totale $\leq C$. Ces patterns sont représentés à l'aide d'une matrice A ($n \times K$).



Par exemple, pour $s = \{18, 19\}$ et $C = 147$, il y a 8 patterns à considérer.

$$A = \begin{pmatrix} 8 & 7 & 6 & 5 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}$$

Cutting Stock Problem - Formulation

Définition formelle :

En utilisant des rouleaux de longueur standard C , on souhaite, en les découpant, di pièces de longueur s_i (avec $0 < s_i \leq C$), $i=1, \dots, n$. Comment organiser ces découpes afin de minimiser le nombre de rouleaux standards utilisé ?

Approche : on génère tous les patterns possibles (toutes les combinaisons de longueur totale $\leq C$. Ces patterns sont représentés à l'aide d'une matrice A ($n \times K$).

NP-Hardness

Les problèmes décrits précédemment sont NP-hard au sens fort.

Prenons par exemple le problème de Bin Packing. Il est possible d'utiliser une réduction du problème de 3-PARTITION.

De retour au Bin Packing – Algorithmes d'approximation

Des algorithmes gloutons efficaces existent pour le Bin Packing :

- First Fit : on traite les objets un par un dans l'ordre, et, pour chacun d'eux, on traite les boîtes (bins) déjà présentes dans la solution, et l'objet est placé dans la première boîte qui convient. Si aucune boîte déjà présente ne convient, une nouvelle boîte est ajoutée (avec l'objet).
- First Fit Decreasing : même chose que l'algorithme précédent, mais les objets sont d'abord triés par ordre décroissant.

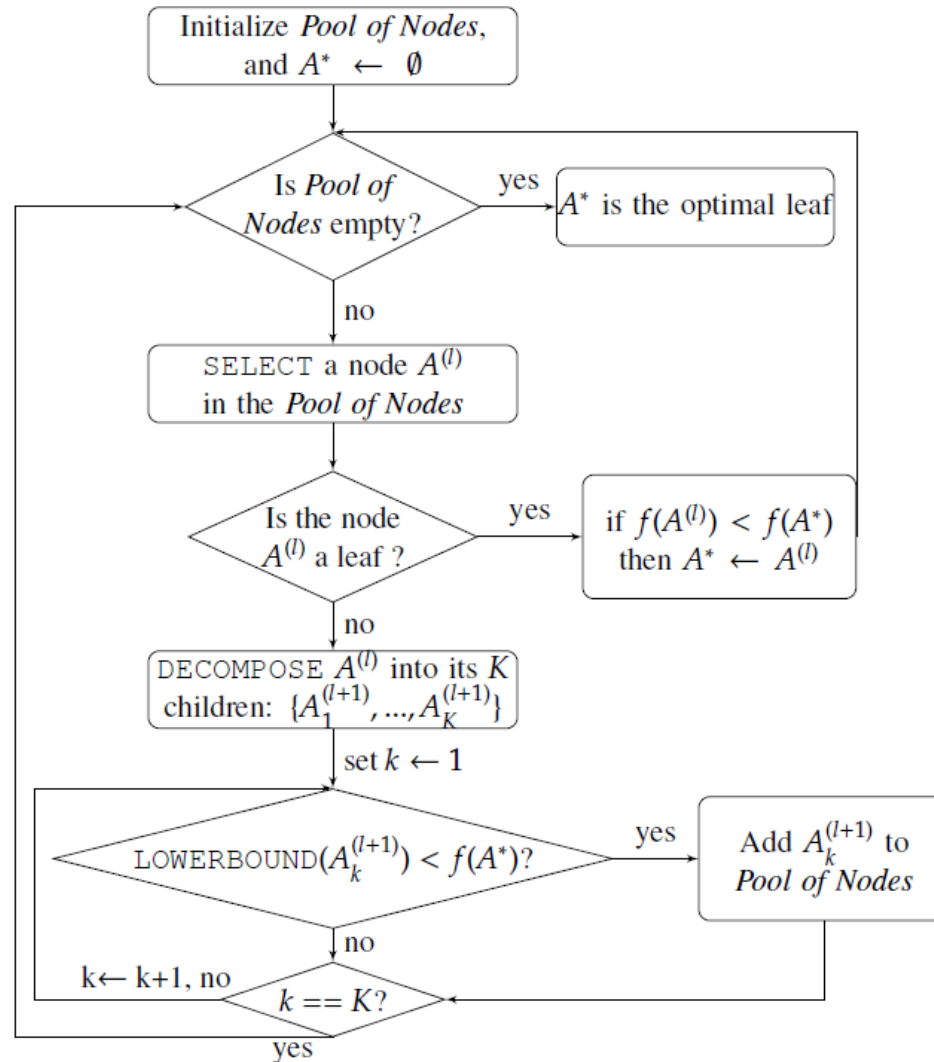
Ce sont des « algorithmes d'approximation » : il est possible de prouver une borne sur l'objectif de la solution générée !

Par exemple, on a pour n'importe quelle instance (s,C) :

$$FF(s,C) \leq 1,7 * OPT(s,C)$$

$$FFD(s,C) \leq (11/9) * OPT(s,C) + 6/9$$

Branch & Bound – Framework général



Comment l'adapter au Bin Packing ?

Branch & Bound – Framework général

Implémenter un Branch & Bound efficace pour un problème donné amène à de nombreuses questions. Il est important de porter une attention particulière aux éléments suivants :

- la borne inférieure : au plus la borne inférieure est efficace, au moins de nœuds seront explorés. Mais le calcul de la borne ne peut pas être trop important !
- solution initiale : une bonne solution initiale permet d'éviter l'évaluation de nombreux nœuds.
- l'ordre d'exploration des solutions partielles : explorer une certaine partie de l'arbre en priorité peut conduire à trouver plus rapidement des bonnes solutions. Par exemple, on explore d'abord les solutions partielles possédant la borne inférieure la plus faible.