

# Continuous optimization

Samuel Buchet & Dorian Dumez & Brendan Guevel

Mai 2017

## 1 Introduction

Dans ce projet, la méthode de la relaxation lagrangienne est utilisée afin d'obtenir des bornes pour le problème de bin packing. Dans un premier temps, une formulation du problème en programme linéaire en nombres entiers est proposée. Ensuite, une heuristique de construction de solution est détaillée, suivi du détail du calcul de la relaxation linéaire. Enfin, plusieurs relaxations lagrangiennes sont proposées et des tests sont interprétés.

## 2 Problème étudié

Le problème de bin packing consiste à insérer des objets de tailles différentes dans des boîtes. Toutes les boîtes ont une même capacité et le but est de minimiser le nombre de boîte nécessaire pour ranger tous les objets. Dans cette version étudiée, le problème ne considère qu'une seule dimension.

On peut écrire le problème de bin-packing monodimensionnel comme :

$$\begin{aligned} \text{(PE)} : z = \min \sum_{b=1}^m u_b \\ \text{s.c.} \quad \sum_{p=1}^n s_p x_{pb} \leq u_b c \quad \forall b \in \{1, \dots, m\} \end{aligned} \tag{1}$$

$$\begin{aligned} \sum_{b=1}^m x_{pb} = 1 \quad \forall p \in \{1, \dots, n\} \\ u_b, x_{pb} \in \{0, 1\} \quad \forall p \in \{1, \dots, n\}, \forall b \in \{1, \dots, m\} \end{aligned} \tag{2}$$

Avec pour paramètres :

- $n$  le nombre d'objet
- $m$  le nombre de bin maximal, peut être obtenu à l'aide d'une heuristique ou par défaut  $n$  (mais sera coûteux)
- $c$  la capacité d'une boîte
- $s$  le vecteur de la taille des objets

Et pour variables :

$$u_b = \begin{cases} 1 & \text{si on utilise le bin numéro } b \\ 0 & \text{sinon} \end{cases}$$

$$x_{pb} = \begin{cases} 1 & \text{si on met l'objet } p \text{ dans le bin } b \\ 0 & \text{sinon} \end{cases}$$

Bien que juste ce programme linéaire n'est pas très utilisable en pratique, en effet :

- il ne prend pas en compte la redondance des objets, si on a 100 fois le même objet il créera 100 variables
- bien qu'identique de part leur caractéristique tous les bin sont différencié

Il en résultera un nombre considérable de solutions équivalentes due à des symétries. Un algorithme de résolution devrait donc prendre en compte l'interprétation du problème ou y ajouter des contraintes comme :

$$u_b \geq u_{b+1} \quad \forall b \in \{1, \dots, m-1\} \quad (3)$$

$$\operatorname{argmin}_{p \in \{i \in \{1, \dots, n\} | x_{ib}=1\}} p \leq \operatorname{argmin}_{p \in \{i \in \{1, \dots, n\} | x_{i,b+1}=1\}} p \quad \forall b \in \{1, \dots, m-1\} \quad (4)$$

Où la contrainte (3) force les bins ouverts à être ceux avec les plus petits identifiants et la contrainte (4) ordonne les bins par le schéma d'objet qu'il contient. Mais même si cela ne règle pas le problème des variables redondantes, et rend ce défaut nécessaire par leur utilisation.

Si on veut les écrire avec des contraintes linéaires, on peut écrire la contrainte (4) avec :

$$x_{pb} \leq \sum_{p=1}^{p-} x_{pp,b-1} \quad \forall p \in \{2, \dots, n\} : \forall b \in \{2, \dots, m\} \quad (4')$$

$$x(1,1) = 1 \quad (4'')$$

De plus on peut remarquer que sous ces contraintes on a toujours :

$$x_{pb} = 0 \quad \forall p \in \{1, \dots, n\} \quad \forall b \in \{p+1, \dots, m\} \quad (5)$$

### 3 Relaxation linéaire

La relaxation linéaire de (PE) s'écrit :

$$(Rl) : z_{Rl} = \min \sum_{b=1}^n u_b$$

$$\text{s.c } \sum_{p=1}^n s_p x_{pb} \leq u_b c \quad \forall b \in \{1, \dots, m\} \quad (1)$$

$$\sum_{b=m}^n x_{pb} = 1 \quad \forall p \in \{1, \dots, n\} \quad (2)$$

$$u_b, x_{pb} \in [0, 1] \quad \forall p \in \{1, \dots, n\} \quad \forall b \in \{1, \dots, m\}$$

De plus on peut montrer que la valeur de la relaxation linéaire est  $\frac{\sum_{p=1}^n s_p}{c}$ . La valeur optimale de (PE) étant toujours entière, peut ainsi utiliser comme borne inférieure  $\left\lceil \frac{\sum_{p=1}^n s_p}{c} \right\rceil$

*Proof.* Soit  $u_b^*$  et  $x_{bp}^*$  la valeur des variables dans la solution optimale de la relaxation linéaire. Alors on a  $\forall p \in \{1, \dots, n\} : \sum_{b=1}^m x_{bp}^* = 1$  par la contrainte (2). De plus, étant donné que chaque  $u_b$  n'apparaît que dans

une contrainte, on peut dire que  $u_b^* = \frac{\sum_{p=1}^n s_p x_{bp}^*}{c}$  par la contrainte (1). En effet ce problème est un problème de minimisation donc cette contrainte est limitante. Par propriété de l'algorithme du simplexe elle sera vérifiée à l'égalité dans la solution optimale (car elles sont toutes indépendantes). La valeur cette solution est alors

$$\sum_{b=1}^m \frac{\sum_{p=1}^n s_p x_{bp}^*}{c} = \frac{\sum_{p=1}^n s_p}{c} \text{ par la contrainte (2).}$$

□

## 4 Heuristique de construction

Afin d'obtenir rapidement une borne supérieurs intéressante pour ce problème, une heuristique de construction peut être utilisée. Dans ce projet, l'heuristique best fit a été implémentée. L'algorithme de best fit consiste à ajouter les objets du plus grand au plus petit. De plus, l'ajout est d'abord tenté dans les bins les plus remplis en premier. Si l'objet ne peut être ajouté dans aucun bin, un nouveau bin est alors ouvert.

## 5 Relaxations lagrangiennes possibles

Plusieurs choix de contraintes étaient possibles pour la relaxation lagrangienne. Les contraintes dualisées ont été choisies afin d'obtenir des sous-problèmes simples à résoudre et décomposable. Dans cette partie, deux relaxations qui nous semblaient intéressantes sont détaillées.

### 5.1 Dualisation de la contrainte 1

Premièrement on remarque que l'on peut séparer en deux la contrainte (1) et donc écrire (PE) comme :

$$\begin{aligned} \text{(PE)} : z = & \min \sum_{b=1}^m u_b \\ \text{s.c.} \end{aligned}$$

$$\forall b \in \llbracket 1, m \rrbracket : \sum_{p=1}^n s_p x_{pb} \leq c \quad (1')$$

$$\forall p \in \llbracket 1, n \rrbracket : \forall b \in \llbracket 1, m \rrbracket : x_{pb} \leq u_b \quad (1'')$$

$$\forall p \in \llbracket 1, n \rrbracket : \sum_{b=1}^m x_{pb} = 1 \quad (2)$$

$$\forall p \in \llbracket 1, n \rrbracket : \forall b \in \llbracket 1, m \rrbracket : u_b, x_{pb} \in \{0, 1\}$$

Alors si on dualise la contrainte (1') on obtient :

$$RL_{1'}(u) : z_{1'}(u) = \min \sum_{b=1}^m u_b - \mu_b \left( c - \sum_{p=1}^n s_p x_{pb} \right)$$

s.c.

$$\forall p \in \llbracket 1, n \rrbracket : \forall b \in \llbracket 1, m \rrbracket : x_{pb} \leq u_b \quad (1'')$$

$$\forall p \in \llbracket 1, n \rrbracket : \sum_{b=1}^m x_{pb} = 1 \quad (2)$$

$$\forall p \in \llbracket 1, n \rrbracket : \forall b \in \llbracket 1, m \rrbracket : u_b, x_{pb} \in \{0, 1\}$$

Où  $\mu \geq 0$  car la contrainte (1') est une contrainte en inégalité.

En écrivant la fonction objectif comme  $z_{1'}(u) = \min \sum_{b=1}^m (u_b + \sum_{p=1}^n (\mu_b s_p x_{pb})) - \sum_{b=1}^m \mu_b c$ . On peut alors voir  $RL_{1'}(u)$  comme un problème d'UFLP (uncapacited facility location problem) avec des coûts d'ouverture de 1 et des coûts d'associations de  $\mu_b s_p$ . A noter que l'on ajoute un terme constant à la fonction objectif pour obtenir la vraie valeur de la relaxation lagrangienne.

Alors comme on l'a fait en cours on peut dualiser la contrainte (2) pour obtenir :

$$RL_{1',2}(u) : z_{1',2}(u) = \min \sum_{b=1}^m (u_b + \sum_{p=1}^n (\mu_b s_p x_{pb})) - \sum_{b=1}^m \mu_b c + \sum_{p=1}^n \gamma_p (1 - \sum_{b=1}^m x_{pb})$$

s.c.

$$\forall p \in \llbracket 1, n \rrbracket : \forall b \in \llbracket 1, m \rrbracket : x_{pb} \leq u_b \quad (1'')$$

$$\forall p \in \llbracket 1, n \rrbracket : \forall b \in \llbracket 1, m \rrbracket : u_b, x_{pb} \in \{0, 1\}$$

Où  $\gamma$  est de signe libre car la contrainte associé est une contrainte d'égalité.

Comme précédemment on peut reformuler la fonction objectif comme :

$$z_{1',2}(u) = \min \sum_{b=1}^m (u_b + \sum_{p=1}^n ((\mu_b s_p - \gamma_p) x_{pb})) - \sum_{b=1}^m \mu_b c + \sum_{p=1}^n \gamma_p$$

En pratique on remarque qu'il suffit de résoudre  $m$  problème indépendants :

$$PI_b(u) : z'_b(u) = \min u_b + \sum_{p=1}^n ((\mu_b s_p - \gamma_p) x_{pb})$$

s.c.

$$\forall p \in \llbracket 1, n \rrbracket : x_{pb} \leq u_b \quad (1'')$$

$$\forall p \in \llbracket 1, n \rrbracket : u_b, x_{pb} \in \{0, 1\}$$

Alors on sait comment calculer la solution optimale de chacun d'entre eux :

$$1 - u_b = 1 \Leftrightarrow \sum_{p \in \llbracket 1, n \rrbracket : \mu_b s_p - \gamma_p < 0} (\mu_b s_p - \gamma_p) < -1$$

$$2 - \forall p \in \llbracket 1, n \rrbracket : (\mu_b s_p - \gamma_p < 0) \Rightarrow x_{pb} = u_b$$

$$3 - \forall p \in \llbracket 1, n \rrbracket : (\mu_b s_p - \gamma_p \geq 0) \Rightarrow x_{pb} = 0$$

De plus on peut remarquer que la contrainte anti-symétrie (5) ne nuit pas au lagrangien : les sous-problèmes reste indépendants et la relaxation conserve sa liberté. Cela permet d'améliorer la vitesse de convergence de l'algorithme de sous-gradient ainsi que la vitesse de résolution d'une partie des sous-problèmes.

## 5.2 Dualisation de la contrainte 2

Une autre possibilité est de dualiser la contrainte (2) :  $\sum_{b=1}^m x_{pb} = 1$ .

On obtient alors :

$$RL_2(u) : z_2(u) = \min \sum_{b=1}^m u_b + \sum_{p=1}^n \mu_p (1 - \sum_{b=1}^m x_{pb})$$

s.c.

$$\begin{aligned} \forall b \in \llbracket 1, m \rrbracket : \sum_{p=1}^n s_p x_{pb} \leq u_b c \quad (1) \\ \forall p \in \llbracket 1, n \rrbracket : \forall b \in \llbracket 1, m \rrbracket : x_{pb}, u_b \in \{0, 1\} \end{aligned}$$

Etant donné que la contrainte (2) est une contrainte d'égalité le signe des  $\mu_i$  est libre.

On peut réécrire la fonction objectif comme  $z_2(u) = \min \sum_{b=1}^m \left( u_b - \sum_{p=1}^n \mu_p x_{pb} \right) + \sum_{p=1}^n \mu_p$ .

Le problème est alors décomposable par bin. En effet, le terme de droite est constant et la somme de gauche peut être décomposée par bin. Les variables d'un terme de la somme ne dépendent que d'un seul bin et il y a aussi une unique contrainte par bin. On se retrouve donc avec les  $m$  sous problèmes identiques :

$$\begin{aligned} \min \quad z_{2b}(0) = u_b - \sum_{p=1}^n \mu_p * x_{pb} \\ \text{s.c. : } \sum_{p=1}^n s_p * x_{pb} \leq u_b * c \end{aligned}$$

Ce problème peut être décomposé en deux sous problèmes : le problème dans lequel  $u_b = 0$  et celui dans lequel  $u_b = 1$ .

Dans le cas où  $u_b = 0$ , la contrainte force les variables  $x_{pb}$  à être nulles (les  $s_p$  étant tous positifs). la valeur optimale de la fonction objectif est alors 0.

Dans le cas où  $u_b = 1$  la contrainte (1) devient une contrainte de capacité. Le problème peut alors être vu comme un problème de sac à dos en inversant le signe de la fonction. Les objets de ce problème ont pour poids  $s_p$  et pour valeurs  $\mu_p$ .

Enfin, la valeur optimal de ces sous problème peut être trouvée en comparant les résultat obtenus avec le problème de sac à dos et le problème avec  $u_b = 0$ . La valeur optimale du premier sous problème étant 0, le deuxième donne une meilleure valeur si  $u_b - \sum_{p=1}^n \mu_p * x_{pb} < 0$  ou encore  $1 < \sum_{p=1}^n \mu_p * x_{pb}$ . On peut donc résoudre le problème en résolvant le sous-problème de sac à dos et en comparant le résultat avec 1.

On peut ainsi résoudre  $RL_2(u)$  en résolvant 1 problème de sac à dos, en multipliant le résultat par  $m$  et en ajoutant la somme des  $\mu$ .

## 6 Algorithme de sous-gradient

Afin de trouver les multiplicateurs des contraintes dualisées de la relaxation qui donnent la meilleure relaxation possible, nous avons implémenté un algorithme de sous-gradient avec une longueur de pas dépendant de la satisfaction des contraintes.

- L'initialisation comprend la mise à 0 des variables de calcul ainsi que la fixation des paramètres, le calcul de borne et l'affectation des valeurs initiale des coefficients lagrangiens
- Toutes les résolutions se font selon des procédures spécifiques à la relaxation choisie, détaillé dans les paragraphes correspondant, de même pour le calcul de la valeur de la fonction objectif
- Le calcul du pas se fait en 2 partie :

---

**Algorithm 1** algorithme de sous-gradient

---

```
1: function RELAXLAGRANGE(instance)
2:   initialisation
3:   première résolution des sous-problèmes
4:   calcul du score et de la première valeur du pas
5:   while  $\neg$  Critères d'arrêts do
6:     calcul des nouveaux multiplicateur
7:     résolution des sous-problèmes
8:     mise à jours de la meilleure solution
9:     calcul de la taille du pas
10:  end while
11:  return best
12: end function
```

---

1 - Le calcul du pas de ce pas :  $\nu = \epsilon \cdot \frac{\omega - val}{\|d - Dx(\mu)\|^2}$  où  $\omega$  est un majorant de la valeur de la relaxation lagrangienne (la valeur de best-fit est utilisée en pratique),  $val$  est la valeur courante de la fonction objectif dualisé, et  $Dx\{\leq, =, \geq\}d$  sont les contraintes qui ont été dualisé ( $x(\mu)$  est la valeur optimale de  $x$  dans les sous-problème avec les coef  $\mu$ ).

2 - Ensuite le pas est  $\mu = \max(\mu + \nu(d - Dx(\mu)), 0)$

$\epsilon$  est un facteur qui va déterminer la taille du pas, au départ il est fixé à une valeur intermédiaire puis va être diminué d'un facteur  $\rho$  si aucune amélioration n'a été effectuée en  $t_{max}$  itérations, avec une ré-augmentation si sa valeur devient vraiment trop faible. De plus, dans le cadre de la première relaxation, on utilise deux  $\nu$  et  $\epsilon$ , un pour chaque type de contraintes. En effet la nature des contraintes dualisées étant profondément différente les mêmes valeurs ne leur conviennent pas. Enfin il faut noter que le signe du multiplicateur est libre ( $\gamma$  de la première est  $\mu$  de la seconde) il n'y a pas de max.

- On dispose de plusieurs critères d'arrêt, l'activation de l'un d'entre eux suffit :
  - $\nu < \nu_{min} \wedge val > \bar{\omega} - 1$  où  $\nu_{min}$  est un seuil sur  $\nu$  et  $\bar{\omega}$  la valeur de la relaxation linéaire. Cette formule signifie que l'algorithme a convergé vers une valeur au moins aussi bonne que la relaxation linéaire (la valeur de la relaxation lagrangienne est sensé être au moins aussi bonne).
  - $\omega - val < 1$  qui signifie que la solution construite est optimale
  - $\nu = 0$  qui signifie que les contraintes dualisées sont toutes vérifiées
  - $no\_improve \geq max\_no\_improve$  qui limite le temps de calcul maximal

## 7 Tests

## 8 Détails sur le programme

## 9 Conclusion