# Bin Packing Robuste

Combalbert Théo - Ametana Komlanvi Parfait

Master ROAD

## 1 Introduction

Nous nous intéressons à la résolution de Variante robuste statique de Bin-Packing avec tailles incertaines. Nous allons utiliser la technique basée sur la dualité de programmation linéaire proposée par [Berstimas et Sim, 2004].On utilise la formulation standard, qui repose sur la connaissance d'un majorant U sur le nombre de boîtes utilisées:

$$\min \sum_{j=1}^{U} y_j \qquad (1)$$

$$\sum_{j=1}^{U} x_{ij} = 1 \qquad \text{i = 1, ..., n} \qquad (2)$$

$$\sum_{i=1}^{n} c_i \, x_{ij} \leq C \, y_j \qquad \text{j = 1, ..., U} \qquad (3)$$

$$x_{ij} \in \{0,1\} \qquad \text{i = 1, ..., n ; j = 1, ..., U} \qquad (4)$$

$$y_j \in \{0,1\} \qquad \text{j = 1, ..., U} \qquad (5)$$

## 2 Reformulations

**Question 1.** Transformation en PL mixte (Berstimas et Sim, 2004).

Dans un premier temps, on remplace chaque contrainte de (2) par sa contrepartie robuste de la manière suivante :

$$\sum_{i=1}^{n} \overline{c_i} \, x_{ij} + max \, \{ \sum_{i=1}^{n} d_i \, x_{ij} \, \xi_i \mid \xi \in \Xi \} \leq C \, y_j \qquad \text{j = 1, ..., U} \qquad (2')$$

où $\Xi = \{ \, \xi \in \{0,1\}^n \mid \sum_{i=1}^{n} \xi_i \leq \Gamma \, \}$ est l'ensemble des sénarios possible ($\xi_i$ =1 si l'article i a une taille différente de $\overline{c_i}$, 0 sinon).

La fonction de protection peut s'écrire sous la forme :
$B_j(x) = max\ \{\sum_{i=1}^{n} d_i\ x_{ij}\ \xi_i\ \xi : \mathrm{A}^t\xi \leq \Gamma$ et $\xi \in \{0,1\}^n\}$ où A est constituée de 1 consécutifs sur chaque ligne. A est donc totalement unimodulaire. Comme $\Gamma \in N$, les points extrêmes de $\{\sum_{i=1}^{n} d_i\ x_{ij}\ \xi_i : \mathrm{A}^t\xi \leq \Gamma$ et $\xi \in [0,1]^n\}$ sont entiers. On peut donc relacher la contrainte d'intégrité de $\xi$ dans $B_j(x)$ et obtenir le Programme Linéaire suivant :

$$B_j(x) = \max \sum_{i=1}^{n} d_i\ x_{ij}\ \xi_i \tag{6}$$

s.c.

$$\sum_{i=1}^{n} \xi_i \leq \Gamma \tag{7}$$

$$\xi_i \leq 1 \qquad i = 1, ..., n \tag{8}$$

$$\xi_i \geq 0 \qquad i = 1, ..., n \tag{9}$$

Ce Programme Linéaire est réalisable borné donc on peut appliquer le théorème de dualité forte. En attribuant la variable $\mu_j$ à la contrainte (7) et les variables $\pi_{ij}$ aux contraintes (8), on obtient le problème dual suivant:

$$B_j(x) = \min \Gamma\ \mu_j + \sum_{i=1}^{n} \pi_{ij}$$

s.c.

$$\mu_j + \pi_{ij} \geq d_i\ x_{ij} \qquad i = 1, ..., n\ ;\ j = 1, ..., U$$

$$\pi_{ij} \geq 0 \qquad i = 1, ..., n\ ;\ j = 1, ..., U$$

$$\mu_j \geq 0 \qquad j = 1, ..., U$$

En utilisant la formulation duale de la fonction de protection dans (2') et en introduisant donc les variables duales $\pi_{ij}$ et $\mu_j$, On obtient le modèle suivant qui est la formulation déterministe équivalente, sous forme de PL mixte de taille polynomiale en la taille des données du problème, de la variante robuste:

$$\min \sum_{j=1}^{U} y_j \tag{10}$$

s.c.

$$\sum_{j=1}^{U} x_{ij} = 1 \qquad \mathrm{i} = 1, ..., \mathrm{n} \tag{11}$$

$$\sum_{i=1}^{n} \overline{c_i}\ x_{ij} + \mu_j\ \Gamma + \sum_{i=1}^{n} \pi_{ij} \leq C\ y_j \qquad \mathrm{j} = 1, ..., \mathrm{U} \tag{12}$$

$$\mu_j + \pi_{ij} \geq d_i\ x_{ij} \qquad \mathrm{i} = 1, ..., \mathrm{n}\ ;\ \mathrm{j} = 1, ..., \mathrm{U} \tag{13}$$

$$x_{ij} \in \{0,1\} \quad \text{i} = 1, ..., \text{n} ; \text{j} = 1, ..., \text{U} \tag{14}$$

$$y_j \in \{0,1\} \quad \text{j} = 1, ..., \text{U} \tag{15}$$

$$\pi_{ij} \geq 0 \quad \text{i} = 1, ..., \text{n} ; \text{j} = 1, ..., \text{U} \tag{16}$$

$$\mu_j \geq 0 \quad \text{j} = 1, ..., \text{U} \tag{17}$$

**Question 2.** Reformulation de Dantzig-Wolfe.

Nous allons appliquer la reformulation de Dantzig-Wolfe à la variante robuste en gardant les contraintes (11), (14) et (15) dans le maître, et une colonne représente un ensemble d'articles affectés à une boîte.

Soit $\text{X}^j = \{\text{x}_{ij} \in \{0,1\},\ \pi_{ij} \geq 0\ ,\ \text{i} = 1, ..., \text{n}\ ,\ y_j \in \{0,1\},\ \mu_j \geq 0 \ :\ \sum_{i=1}^{n} \overline{c_i}$
$x_{ij} + \mu_j\ \Gamma + \sum_{i=1}^{n} \pi_{ij} \leq C,\ \ \mu_j + \pi_{ij} \geq d_i\ x_{ij},\quad \text{i} = 1, ..., \text{n}\ \}$ (18)

Soit $P^j$ l'ensemble des points entiers de $X^j$.

On introduit les variables $\lambda_p^j (j = 1, ..., U;\ \text{p} \in P^j)$

$$\lambda_p^j = \begin{cases} 1 \text{ si on utilise la solution p} \in P^j \\ 0 \text{ sinon} \end{cases}$$

La solution p est décrite par $\overline{x_{ij}^p},\ \overline{y_j^p},\ \overline{\pi_{ij}^p}\ et\ \overline{\mu_j^p}$ les valeurs de $x_{ij},\ y_j,\ \pi_{ij},\ \mu_j$ correspondants à cette solution.

On obtient la reformulation suivante:

$$\min \sum_{j=1}^{U} y_j \tag{19}$$

$$\sum_{j=1}^{U} x_{ij} = 1 \quad \text{i} = 1, ..., \text{n} \tag{20}$$

$$x_{ij} = \sum_{p \in P^j} \overline{x_{ij}^p} \lambda_p^j \quad \text{i} = 1,..., \text{n} ; \text{j}= 1,...,\text{U} \tag{21}$$

$$y_j = \sum_{p \in P^j} \overline{y_j^p} \lambda_p^j \quad \text{j}= 1,...,\text{U} \tag{22}$$

$$\sum_{p \in P^j} \lambda_p^j = 1 \quad \text{j}= 1,...,\text{U} \tag{23}$$

$$\lambda_p^j \geq 0 \quad j = 1, ..., U; p \in P^j \tag{24}$$

$$\mathrm{x}_{ij} \in \{0,1\} \quad \mathrm{i} = 1, ..., \mathrm{n}\ ; \mathrm{j} = 1, ..., \mathrm{U} \tag{25}$$

$$y_j \in \{0,1\} \quad \mathrm{j} = 1, ..., \mathrm{U} \tag{26}$$

En éliminant les variables $x_{ij}$ et $y_j$ on obtient:

**le Problème maître:**

$$\min \sum_{j=1}^{U} \sum_{p \in P^j} \overline{y_j^p} \lambda_p^j \tag{27}$$

s.c.

$$\sum_{j=1}^{U} \sum_{p \in P^j} \overline{x_{ij}^p} \lambda_p^j = 1 \quad \mathrm{i} = 1, ..., \mathrm{n} \tag{28}$$

$$\sum_{p \in P^j} \lambda_p^j = 1 \quad \mathrm{j} = 1, ..., \mathrm{U} \tag{29}$$

$$\lambda_p^j \geq 0 \quad j = 1, ..., U; p \in P^j \tag{30}$$

**le sous-problème (pour j):**

Soient $\theta_j$ la variable duale associée à la contrainte (29) et les $\alpha_i$ les variables duale associés aux contraintes (28)

$$\min y_j - \theta_j - \sum_{i=1}^{n} \alpha_i x_{ij} \tag{31}$$

sc:

$$\sum_{i=1}^{n} \overline{c_i}\ x_{ij} + \mu_j\ \Gamma + \sum_{i=1}^{n} \pi_{ij} \leq C \tag{32}$$

$$\mu_j + \pi_{ij} \geq d_i\ x_{ij} \quad \mathrm{i} = 1, ..., \mathrm{n} \tag{33}$$

$$x_{ij} \in \{0,1\} \quad \mathrm{i} = 1, ..., \mathrm{n} \tag{34}$$

$$y_j \in \{0,1\} \tag{35}$$

$$\pi_{ij} \geq 0 \quad \mathrm{i} = 1, ..., \mathrm{n} \tag{36}$$

$$\mu_j \geq 0 \tag{37}$$

On remarque que tous les sous-problèmes sont identiques (pour tous les j). Si on résout un sous-problème, on résout tous les sous problèmes.
On va donc agréger les sous-problèmes, pour obtenir le modèle suivant, en supprimant l'indice j qui devient redondant.

$$\min \sum_{p \in P} \overline{y^p} \lambda_p \tag{38}$$

$$\sum_{p \in P} \overline{x_{ij}^p} \lambda_p = 1 \qquad \text{i} = 1, ..., \text{n} \tag{39}$$

$$\sum_{p \in P} \lambda_p = \text{U} \tag{40}$$

$$\lambda_p \geq 0 \qquad p \in P \tag{41}$$

De plus, on remarques également que si $\overline{y^p} = 0$ alors le sac n'est pas utilisé. on peut donc simplifier le maître en ne comptant seulement que le nombre de solutions non vides utilisées.

On pose $\overline{P} = \{\text{x}_i \in \{0,1\},\ \pi_i \geq 0\ ,\ \text{i} = 1, ..., \text{n}\ ,\ \ \mu \geq 0\ \ :\ \sum_{i=1}^{n} \overline{c_i}\ x_i + \mu\ \Gamma + \sum_{i=1}^{n} \pi_i \leq C,\ \ \mu + \pi_i \geq d_i\ x_i,\quad \text{i} = 1, ..., \text{n}\ \}$ (42)

Nous obtenons ainsi
**le nouveau Problème maître:**

$$\min \sum_{p \in \overline{P}} \lambda_p \tag{43}$$

sc:

$$\sum_{p \in \overline{P}} \overline{x_i^p} \lambda_p = 1 \qquad \text{i} = 1, ..., \text{n} \tag{44}$$

$$\lambda_p \geq 0 \qquad p \in \overline{P} \tag{45}$$

**Question 3.** Problème de pricing.

Soient $\alpha_i$ les variables duales associés aux contraintes (44)

$$\min 1 - \sum_{i=1}^{n} \alpha_i x_i \tag{46}$$

sc:

$$\sum_{i=1}^{n} \overline{c_i}\ x_i + \mu\ \Gamma + \sum_{i=1}^{n} \pi_i \leq C \tag{47}$$

$$\mu + \pi_i \geq d_i\ x_i \qquad \text{i = 1, ..., n} \tag{48}$$

$$x_i \in \{0,1\} \qquad \text{i = 1, ..., n} \tag{49}$$

$$\pi_i \geq 0 \qquad \text{i = 1, ..., n} \tag{50}$$

$$\mu \geq 0 \tag{51}$$

Ce Programme Linéaire est équivalent à :

$$\max \sum_{i=1}^{n} \alpha_i x_i \tag{52}$$

sc:

$$\sum_{i=1}^{n} \overline{c_i}\ x_i + \mu\ \Gamma + \sum_{i=1}^{n} \pi_i \leq C \tag{53}$$

$$\mu + \pi_i \geq d_i\ x_i \qquad \text{i = 1, ..., n} \tag{54}$$

$$x_i \in \{0,1\} \qquad \text{i = 1, ..., n} \tag{55}$$

$$\pi_i \geq 0 \qquad \text{i = 1, ..., n} \tag{56}$$

$$\mu \geq 0 \tag{57}$$

Notre problème de problème de pricing est une variante du Sac à dos, donc on peut le résoudre par programmation dynamique.

Nous allons supposer que les articles sont triés par ordre croissant des $d_i$. A chaque fois qu'on se trouve sur un arcticle i, il sera question de soit "ne pas le prendre"ou "le prendre avec son poids maximal$(c_i + d_i)$" ou encore "le prendre avec son poids norminal$c_i$. Nous allons donc resoudre deux sous-programme dynamiques.

$1^{ère}$ récurrence:
$P(i, c, \gamma) = max\{P(i-1, c, \gamma); P(i-1, c - c_i - d_i, \gamma - 1) + p_i\}$, c=0,...,C ; $\gamma$=

1,...,Γ ; i=1,...,n

$2^{ième}$ récurrence:
$Q(i,c) = max\{Q(i-1,c); Q(i-1,c-c_i)+p_i\}$, c=0,...,C ; i=Γ+1,...,n

Initialisation:
-$P(0,c,\gamma) = -\infty$ c=0,...,C ; $\gamma$= 1,...,Γ
-$P(0,0,0) = 0$ c=0,...,C ; $\gamma$= 1,...,Γ
-$Q(\Gamma,c) = P(\Gamma,c,\Gamma)$ c=0,...,C

L'optimum est donné par:

$$opt = max \begin{cases} max\ \{\text{Q(n,c)}\ , & \text{c= 1,...,C }\} \\ max\ \{\text{P(n,c,}\gamma\text{)}, & c = 1,...,C; \gamma = 1,...,\Gamma - 1\} \end{cases}$$

**Question 5.** Cas particulier de la variante robuste ($d_i = d$).

Pour modéliser le cas particulier de la variante robuste du sac à dos où $d_i = d$ $i = 1,...,n$, nous utiliserons:
-$\{1...U_1\}$ : ensemble des boites de capacité C
-$\{1...U_2\}$ : ensemble des boîtes de capacité C-Γd
-les variables $x_{ij}$: 1 si l'article i est affecté à la boîte j, 0 sinon , $i = 1,...,n; j = 1,...,U_1$
-les variables $t_{ij}$: 1 si l'article i est affecté à la boîte j, 0 sinon , $i = 1,...,n; j = 1,...,U_2$
-les variables $y_j$: 1 si la boîte j est utilisée, 0 sinon , $ij = 1,...,U_1$
-les variables $z_j$: 1 si la boîte j est utilisée, 0 sinon , $ij = 1,...,U_2$

On obtient donc le modèle suivant:

$$\min \sum_{j=1}^{U_1} y_j + \sum_{j=1}^{U_2} z_j \tag{58}$$

s.c.

$$\sum_{j=1}^{U_1} x_{ij} + \sum_{j=1}^{U_2} t_{ij} = 1 \qquad \text{i = 1, ..., n} \tag{59}$$

$$\sum_{i=1}^{n} (\overline{c_i}+\text{d}_i)\text{x}_{ij} \leq C\ y_j \qquad \text{j = 1, ..., } U_1 \tag{60}$$

$$\sum_{i=1}^{n} x_{ij} \leq \Gamma y_j \qquad \text{j = 1, ..., } U_1 \tag{61}$$

$$\sum_{i=1}^{n} \overline{c_i}\, t_{ij} \leq (C - \Gamma d)\, z_j \qquad \text{j} = 1, ..., U_2 \tag{62}$$

$$x_{ij} \in \{0,1\} \qquad \text{i} = 1, ..., \text{n} \ ; \text{j} = 1, ..., U_1 \tag{63}$$

$$y_j \in \{0,1\} \qquad \text{j} = 1, ..., U_1 \tag{64}$$

$$t_{ij} \in \{0,1\} \qquad \text{i} = 1, ..., \text{n} \ ; \text{j} = 1, ..., U_2 \tag{65}$$

$$z_j \in \{0,1\} \qquad \text{j} = 1, ..., U_2 \tag{66}$$

Nous allons appliquer la reformulation de Dantzig-Wolfe à ce modèle en gardant les contraintes (59), (63) à (66) dans le maître, et une colonne représente un ensemble d'articles affectés à une boîte.

Soit $\text{X}^j = \{\text{x}_{ij} \in \{0,1\},\ \text{y}_j \in \{0,1\} \ : \sum_{i=1}^{n} (\overline{c_i} + d_i)\ x_{ij} \leq C\ ,\ \sum_{i=1}^{n} x_{ij} \leq \Gamma\ \}$, j=1,...,$U_1$ (67)

Soit $\text{T}^j = \{\text{t}_{ij} \in \{0,1\},\ \text{z}_j \in \{0,1\} \ : \sum_{i=1}^{n} \overline{c_i}\, t_{ij} \leq C - \Gamma d\ \}$, j=1,...,$U_2$ (68)

Soientt $P^j$ l'ensemble des points entiers de $X^j$ et $Q^j$ l'ensemble des points entiers de $T^j$.

En utilisant le même raisonnemnt qu'à la question 2 à chaque sous groupe de boîte, il est claire qu'on a deux sous-groupe de sous problème identiques. On peut donc utiliser les résultats précédents, en notant:

-$\overline{P} = \{\text{x}_i \in \{0,1\},\ \ i = 1, ..., n,\ :\ \sum_{i=1}^{n} (\overline{c_i} + d)\, x_i \leq C,\ \ \sum_{i=1}^{n} x_i \leq \Gamma\ \}$ (69)

-$\overline{Q} = \{\text{t}_i \in \{0,1\},\ \ i = 1, ..., n,\ :\ \sum_{i=1}^{n} \overline{c_i}\, t_i \leq C - \Gamma d\ \}$ (70)

On utilise les variables suivantes:

$$\lambda_p = \begin{cases} 1 \text{ si on utilise la solution p} \in \overline{P} \\ 0 \text{ sinon} \end{cases}$$

$$\lambda_q = \begin{cases} 1 \text{ si on utilise la solution q} \in \overline{Q} \\ 0 \text{ sinon} \end{cases}$$

Toute solution $p \in \overline{P}$ est décrite par $\overline{x_i^p}$ tandis que qu'une solution $q \in \overline{Q}$ est décrite par $\overline{t_i^q}$. Sachant qu'on ne compte que les boîte utiliée dans l'objectif, on a :

**Le Programme Maître:**

$$\min \sum_{p \in \overline{P}} \lambda_p + \sum_{q \in \overline{Q}} \lambda_q \tag{71}$$

sc:

$$\sum_{p \in \overline{P}} \overline{x_i^p} \lambda_p + \sum_{q \in \overline{Q}} \overline{t_i^q} \lambda_q = 1 \qquad \text{i = 1, ..., n} \tag{72}$$

$$\lambda_p \geq 0 \qquad p \in \overline{P} \tag{73}$$

$$\lambda_q \geq 0 \qquad q \in \overline{Q} \tag{74}$$

Soient $\alpha_i$ les variables duales associés aux contraintes (72)

**Le Sous Problème 1:**

$$\min \text{ 1-} \sum_{i=1}^{n} \alpha_i x_i = \max \sum_{i=1}^{n} \alpha_i x_i \tag{75}$$

sc:

$$\sum_{i=1}^{n} (\overline{c_i} + d)\, x_i \leq C \tag{76}$$

$$\sum_{i=1}^{n} x_i \leq \Gamma \tag{78}$$

$$x_i \in \{0,1\} \qquad \text{i = 1, ..., n} \tag{79}$$

**Le Sous Problème 2:**

$$\min \text{ 1-} \sum_{i=1}^{n} \alpha_i t_i = \max \sum_{i=1}^{n} \alpha_i t_i \tag{80}$$

sc:

$$\sum_{i=1}^{n} \overline{c_i}\, t_i \leq C - \Gamma d \tag{81}$$

$$t_i \in \{0,1\} \qquad \text{i = 1, ..., n} \tag{82}$$

**Question 6.** Cas entier

Une tentative de branch and price a été implementée avec branchement sur

les variables $x_{ij}$ dans le maître mais sans résultats obtenus. Sachant que $x_{ij}$ $=\overline{x_i^p}\ \lambda_p$ pour un p donné alors peut alors confondre $x_{ij}$ et $x_{ip}$ en associant une configuration a un bin

**Algorithme** :
1. On résoud la relaxation linéaire avec la génération de colonne (recuperer la solution dans LB
2. On récupère les solutions
3. Si que des $x_{ip}$ entiers alors STOP
4. Sinon repérer un $x_{ip}$ à brancher (le plus proche de 0 et non nul)
5. Créer 2 noeuds( $\overline{x_j^p}\ \lambda_p \leq 0$ et $\overline{x_j^p}\lambda_p \geq 1$)
6. Choisir un des 2 noeuds et ajouter la contrainte de brachement au maître courant
7. Marquer ce noeud et Résoudre la la relaxation linéaire avec la génération de colonne en ce noeud
8. Résoudre la relaxation linéaire avec la génération de colonne ( récupérer la solution dans Sol)
9. Si Sol$\leq$ LB ou Sol$\geq$UB alors se déplacer sur un noeud non marqué, le marquer puis aller en (8)
10. Sinon mettre aller en (2)

Impact sur l'objectif du sous problème :
Lorsqu'on ajoute une branche à gauche $\leq 0$ : $\max \sum_{i=1}^{n} \alpha_i x_i + \mu\ x_j$
Lorsqu'on ajoute une branche à droite $\geq 1$ : $\max \sum_{i=1}^{n} \alpha_i x_i$ - $\mu\ x_j$

# 3 Travail d'expérimentations

Dans cette section nous allons comparer les temps de calculs des différents algorithmes ainsi que les solutions trouvées. Un temps limite a été fixé à 300 secondes (5 min). Les implémentations ont été réalisées dans le langage C++ avec pour support l'utilisation du solveur CPLEX. Les tests ont été effectuées sur un ordinateur possédant un processeur Intel® Core™ i5-8365U CPU @ 1.60GHz × 8 , 16 Go de mémoire vive et ayant comme système d'exploitation Ubuntu 20.04.1 LTS.

**Signification des notations utilisées:**

-Model de base : PLNE de Formulation Déterministe équivqlente de la variante robuste avec $d_i$ quelconque

-Relax model de base : Relaxation linéaire de Formulation Déterministe équivqlente de la variante robuste avec $d_i$ quelconque

-Relax gcol quel : Génération de Colonne avec $d_i$ quelconque

-sol entière gcol quel : Branch and price avec $d_i$ quelconque

-Relax gcol iden : Génération de Colonne avec $d_i$ identique

-sol entière gcol iden : Branch and price avec $d_i$ identique

-T.L. : Temps Limite

-objectif : valeur de la fonction objectif ("-" signifie qu'on a pas trouvé de solution)

-temps : Temps d'exécution en secondes

## 3.1 Problème à $d_i$ quelconque

| $\Gamma = 0$ | Model de base | Relax model de base | Relax gcol quel | sol entière gcol quel |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 38(T.L.) | 36.41(0.033) | 37.2031(3.763) | |
| r100_1000_1 | 1000(T.L.) | 367.72(6.08) | 378.051(T.L.) | |
| t100_100_1 | 35(T.L.) | 34(0.04) | 34(3.898) | |
| t100_1000_1 | 1002(T.L.) | 334(6.32) | 334.004(T.L.) | |

| $\Gamma = 1$ | Model de base | Relax model de base | Relax gcol quel | sol entière gcol quel |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 60(T.L.) | 36.48(0.1) | 41.3529(4.062) | |
| r100_1000_1 | -(T.L.) | -(T.L.) | 421.1(105.407) | |
| t100_100_1 | 39(T.L.) | 34.04(0.09) | 36.2714(3.948) | |
| t100_1000_1 | 1002(T.L.) | 334.04(154.98) | 355.842(294.812) | |

| $\Gamma = 2$ | Model de base | Relax model de base | Relax gcol quel | sol entière gcol quel |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 44(T.L.) | 36.54(0.16) | 41.3529(4.909) | |
| r100_1000_1 | -(T.L.) | -(T.L.) | 421.1(115.112) | |
| t100_100_1 | 38(T.L.) | 34.08(157.15) | 36.2714(5.938) | |
| t100_1000_1 | -(T.L.) | 334.08(157.15) | 355.878(T.L.) | |

| $\Gamma = 5$ | Model de base | Relax model de base | Relax gcol quel | sol entière gcol quel |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 45(T.L.) | 36.72(0.09) | 41.3529(3.151) | |
| r100_1000_1 | -(T.L.) | -(T.L.) | 421.1(113.392) | |
| t100_100_1 | 41(T.L.) | 34.17(0.08) | 36.2714(4.762) | |
| t100_1000_1 | -(T.L.) | 334.2(157.7) | 355.889(T.L.) | |

| $\Gamma = 10$ | Model de base | Relax model de base | Relax gcol quel | entière gcol |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 47(T.L.) | 37.01(0.12) | 41.3529(5.290) | |
| r100_1000_1 | -(T.L.) | 368.41(206.88) | 421(118.807) | |
| t100_100_1 | 39(T.L.) | 34.32(0.08) | 36.2714(4.762) | |
| t100_1000_1 | -(T.L.) | 334.4(156.91) | 355.889(T.L.) | |

On remarque que la difficulté du "modele de base " à trouver une solution tandis que sa relaxation linéaire fourni une borne inférieure sur la nombre de bins optimale. Cette borne est ammélioréé par la la relaxation linéaire calculée par génération de colonnes qui nécessite plus de temps. Cette borne est utile dans l'implementatation du Branch and and price pour pour obtenir une solution entière à partir de la génération de colonnes. Dans le cas où le "model de base" fourni un resultat dans le temps limite, ce résultat sert également de borne supérieur pour le branch and price. Nous ne pouvons malheureusement pas faire de comparaison avec les resultats du branch and price, car l'implémentation n'a pas été une réussite.

### 3.2 Problème avec $d_i = d$

| $\Gamma = 0$ | Model de base | Relax model de base | Relax gcol quel | Relax gcol iden |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 38(T.L.) | 36.41(0.033) | 37.2031(3.763) | 37.2031 (9.646) |
| r100_1000_1 | 1000(T.L.) | 367.72(6.08) | 378.051(T.L.) | 378.051(T.L) |
| t100_100_1 | 35(T.L.) | 34(0.04) | 34(3.898) | 34(3.080) |
| t100_1000_1 | 1002(T.L.) | 334(6.32) | 334.004(T.L.) | 334(219.336) |

| $\Gamma = 1$ | Model de base | Relax model de base | Relax gcol quel | Relax gcol iden |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 60(T.L.) | 36.48(0.1) | 41.3529(4.062) | 38.0435(3.519) |
| r100_1000_1 | -(T.L.) | -(T.L.) | 421.1(105.407) | 390.312(230.850) |
| t100_100_1 | 39(T.L.) | 34.04(0.09) | 36.2714(3.948) | 34.1791(8.051) |
| t100_1000_1 | 1002(T.L.) | 334.04(154.98) | 355.842(294.812) | 334.659(T.L.) |

| $\Gamma = 2$ | Model de base | Relax model de base | Relax gcol quel | Relax gcol iden |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 44(T.L.) | 36.54(0.16) | 41.3529(4.909) | 40.5833(1914) |
| r100_1000_1 | -(T.L.) | -(T.L.) | 421.1(115.112) | 399.105(217.347) |
| t100_100_1 | 38(T.L.) | 34.08(157.15) | 36.2714(5.938) | 34.2727(9.922) |
| t100_1000_1 | -(T.L.) | 334.08(157.15) | 355.878(T.L.) | 334.927(T.L.) |

| $\Gamma = 5$ | Model de base | Relax model de base | Relax gcol quel | Relax gcol iden |
|---|---|---|---|---|
| Instances | objectif (temps) | objectif (temps) | objectif (temps) | objectif (temps) |
| r100_100_1 | 45(T.L.) | 36.72(0.09) | 41.3529(3.151) | 41.25(1.579) |
| r100_1000_1 | -(T.L.) | -(T.L.) | 421.1(113.392) | 404.971(273.057) |
| t100_100_1 | 41(T.L.) | 34.17(0.08) | 36.2714(4.762) | 34.3636(15.839) |
| t100_1000_1 | -(T.L.) | 334.2(157.7) | 355.889(T.L.) | 335(T.L.) |

Lorsque les déviations sont identiques et que $\Gamma = 0$, on obtient les mêmes résultats que dans le cas des déviations identique. Cela s'explique par le fait qu'la taille d'aucun arcticle ne subira de déviation et donc on se retrouve avec un problème de Bin-Parking detarministe. Du Coup les variations apportées aux $d_i$ n'impactent pas les solutions, ni les temps de résolution. Par contre lorsque $\Gamma$ est différent de 0, la relaxation linéaire calculée par génération de colonnes avec le modèle avec $d_i$ identique utilise moins de bins que celle avec $d_i$ quelconque. Si on remarque que la relaxation linéaire calculée par génération de colonnes avec le modèle avec $d_i$ identique met plus de temps, cela s'explique par la résolution de deux sous-problèmes. Tout comme dans le cas des $d_i$ quelconque, la solution fourni par cette relaxation sert de borne inférieur dans la résolution du branch and price pour obtenir des solutions entiéres.