Conception d'instances difficiles pour des problèmes multi-dimensionnels de sac-à-dos

V. BOYER, D. EL BAZ, M. ELKIHEL & J.B. LASSERRE

Université de Toulouse
LAAS-CNRS - Laboratoire d'Analyse et d'Architecture des Systèmes
7 avenue du Colonel Roche 31077 Toulouse Cedex 4 France
vboyer@laas.fr, elbaz@laas.fr, elkihel@laas.fr & lasserre@laas.fr

Résumé Le problème du sac-à-dos multi-dimenssionnel est un problème classique de l'optimisation combinatoire appartenant à la classe des problèmes NP-complets. Il fait l'objet d'un grand intérêt de la part de la communauté scientifique se traduisant par une multitude de travaux autour du problème du sac-à-dos. Cependant, il existe peu de travaux portant sur ce qui fait la nature de la difficulté en optimisation combinatoire.

Dans cette article nous abordons le thème de la difficulté par l'étude de diverses techniques permettant de concevoir des instances ardues. Ces dernières sont basées respectivement sur la conception de problèmes de sac-à-dos dérivés de problèmes en contraintes égalité et sur l'utilisation de la transformée en $\mathbb{Z}$ du problème du sac-à-dos multi-dimenssionnel.

Mots-Clefs. Problème du sac-à-dos ; Relaxation Lagrangienne ; Transformée en $\mathbb{Z}$.

1 Introduction

Le problème du sac-à-dos est un problème classique de l'optimisation combinatoire appartenant à la classe des problèmes NP-complets [5]. Il peut être formulé comme suit :

$$(MKP)\begin{cases} \max \sum_{j=1}^{n} p_j.x_j = v(MKP), \\ \text{s.c. } \sum_{j=1}^{n} w_{i,j}.x_j \le c_i,\ i \in \{1,...,m\}, \\ x_j \in \{0,1\},\ j \in \{1,...,n\}, \end{cases} \tag{1}$$

avec p_j, c_i, et $w_{i,j}$ entiers positifs pour $i \in \{1,...,m\}$ et $j \in \{1,...,n\}$.

Ou plus simplement :

$$(MKP)\begin{cases} \max p^t.x, \\ \text{s.c. } W.x \le c, \\ x \in \{0,\ 1\}^n, \end{cases} \tag{2}$$

avec $p = \{p_1, ..., p_n\}^t$, $c = \{c_1, ..., c_m\}^t$ et $W = \begin{pmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{m,1} & \cdots & w_{m,n} \end{pmatrix}$.

De plus, nous nous placerons sous les hypothèses (h1) et (h2) qui écartent les solutions triviales du problème :

– (h1) : $\max_{j \in \{0,...,n\}} \{w_{i,j}\} \le c_i$ pour $i \in \{1,...,m\}$ (cas de l'exclusion d'une variable) ,

– (h2) : $\forall i \in \{0,...,m\}$, $c_i \ge \sum_{j=1}^{m} w_{i,j}$ (cas d'une contrainte toujours réalisée).

On rencontre le problème du sac-à-dos multidimensionnel lorsqu'on cherche à optimiser l'usage d'une entité qui consomme des ressources multiples. Il correspond au modèle général de type allocation de ressources.

Il fait l'objet d'un grand intérêt de la part de la communauté scientifique, souligné par la création de groupe de travail dédié tel que KSO (KnapSack et Optimisation) du GPR RO. Les applications dans ce domaine sont en effet très nombreuses (problèmes liés à la cryptographie, à la logistique, à la productique, à l'économie, au multimédia et à la finance) :

– Gestion des chaînes logistiques (problème de répartition)
– Ordonnancement
– Gestion des ressources humaines (emploi du temps)
– Maintenance (service après vente, ordre des visites, minimisation de parcours)
– Gestion des sièges de trains, avions en fonction des tarifs
– Gestion de ressources (départ, arrivée, placement des avions)
– Gestion des containers dans les ports

De ce fait, la communauté scientifique porte un grand intérêt à la résolution d'instances difficiles et industrielles. Nous nous intéressons ici à ce qui fait qu'un problème est difficile ou non. On sait que dans le cas à une contrainte, plus les ratios profit sur poids sont proches, plus le problème est difficile à résoudre. En se basant sur ce constat, nous avons essayé d'engendrer des instances difficiles dans le cas multicontraint. Deux approches différentes sont présentées.

2 Instances engendrées aléatoirement

La génération d'instances de façon aléatoire soulève la question de ce qui fait qu'une instance est plus ou moins difficiles à résoudre. Peu d'études ont été faites sur ce sujet.

Bien entendu, la difficulté de résolution d'un problème dépend de la taille de ce dernier, mais pas que de cela. On se rend compte qu'en considérant des tests sur des problèmes aléatoires, certains sont beaucoup plus durs à résoudre que d'autres qui peuvent notamment avoir une taille plus grande.

Dans cette partie nous présenterons trois classes de problèmes couramment utilisés afin de tester les algorithmes de résolution, à savoir :
– problèmes à données non-corrélées,
– problèmes à données faiblement corrélées et
– problèmes à données fortement corrélées

2.1 Problèmes non-corrélés

Lorsque l'on étudie la complexité des méthodes de résolutions exacte, on se rend qu'elle est liée d'une part à sa taille, mais aussi à la capacité du problème. Nous pouvons considérer les problèmes suivants :

$$(MKP(\alpha)) \begin{cases} \max \sum_{j=1}^{n} p_j.x_j, \\ \text{s.c.} \sum_{j=1}^{n} w_{i,j}.x_j \leq \alpha.\sum_{j=1}^{n} w_{i,j},\ i \in \{1,...,m\}, \\ x_j \in \{0,1\},\ j \in \{1,...,n\}, \end{cases} \tag{3}$$

avec $\alpha \in]0,1[$ et pour $j \in \{1,...,n\}$ et $i \in \{1,...,m\}$, les coefficients p_j, c_i et $w_{i,j}$ sont tirés de manière uniforme dans, respectivement, $[0,U_p]$, $[0,U_c]$ et $[0,U_w]$ où α, U_p, U_c et U_w sont des paramètres fixés par l'utilisateur.

Ceci constitue la méthode la plus classique pour engendrer des problèmes de façon aléatoire ; elle est d'ailleurs à la base de la génération des instances de Chu et Beasley [3] pour lesquelles α prend les valeurs 0,25, 0,50 et 0,75.

Les instances les plus difficiles sont crées pour $\alpha = 0,5$. En effet, pour un α petit, le sac-à-dos est vite saturé, et pour un α grand, un changement de variable de type $x \rightarrow 1 - x$ permet de se ramener au cas précédent.

2.2 Problèmes corrélés

Contrairement au cas précédent, la génération du vecteur des profits p est liée à la matrice des poids W. On distingue les problèmes :

- faiblement corrélés : pour $j \in \{1, ..., n\}$ et $i \in \{1, ..., m\}$, les coefficients $w_{i,j}$ sont tirés de manière uniforme dans $[0, U_w]$ et $p_j = \max\{1,\ \hat{p}_j\}$ avec $\hat{p}_j$ tiré de manière uniforme dans :

$$\left[\frac{\sum_{k=1}^{m} w_{k,j}}{m} - wc, \frac{\sum_{k=1}^{m} w_{k,j}}{m} + wc \right],$$

  où wc est un paramètre spécifié par l'utilisateur ;
- fortement corrélés : pour $j \in \{1, ..., n\}$ et $i \in \{1, ..., m\}$, les coefficients $w_{i,j}$ sont tirés de manière uniforme dans $[0, U_w]$ et

$$p_j = \frac{\sum_{k=1}^{m} w_{k,j}}{m} + sc,$$

  où sc est un paramètre spécifié par l'utilisateur.

Quant au vecteur des capacités c, il est généralement engendré de la même façon qu'avec les problèmes non-corrélés, à savoir $c = \alpha . \sum_{j=1}^{n} W^j$.

Les problèmes fortement corrélés sont généralement plus difficiles à résoudre que leurs homologues non-corrélés (cf. [15], [13], [16]).

3 Problèmes dérivés d'un problème combinatoire en contrainte égalité

Cette méthode se base sur les travaux de El Baz, Elkihel et Gely [4] qui ont présenté une méthode de résolution du problème du sac-à-dos multidimensionnel en contrainte égalité basée sur la résolution d'une série de problèmes en contrainte inégalité.

On considère les problèmes équivalents :

$$(E) \begin{cases} \max p^t.x, \\ \text{s.c. } W.x = c, \\ x \in \{0,\ 1\}^n, \end{cases} \Leftrightarrow (E) \begin{cases} \max p^t.x, \\ \text{s.c. } \begin{cases} W.x \leq c, \\ W.x \geq c, \end{cases} \\ x \in \{0,\ 1\}^n. \end{cases} \tag{4}$$

Une relaxation Lagrangienne sur la contrainte $W.x \geq c$ donne alors :

$$(\widetilde{E}(\lambda)) \begin{cases} \max p^t.x + \lambda^t.(W.x - c), \\ \text{s.c. } W.x \leq c, \\ x \in \{0,\ 1\}^n, \end{cases} \tag{5}$$

où $\lambda \geq 0$.

Pour un λ^* suffisamment grand on a $v(\widetilde{E}(\lambda^*)) = v(E)$ [4], autrement dit il y a équivalence entre E et $\widetilde{E}(\lambda^*)$. Ceci nous mène donc à engendrer des problèmes de la forme de $\widetilde{E}(\lambda)$ sachant que plus le paramètre λ sera grand, plus le problème relâché est proche du problème en contrainte égalité. En prenant $c = \alpha.W.\overline{x}$, on obtient d'une façon générale des problèmes de la forme :

$$(MKP(\alpha, \lambda)) \begin{cases} \max p^t.x + \lambda^t.(W.x - \alpha.W.\overline{x}), \\ \text{s.c. } W.x \leq \alpha.W.\overline{x}, \\ x \in \{0,\ 1\}^n, \end{cases} \tag{6}$$

avec $\overline{x}^t = (1, ..., 1)$, $\alpha \in]0, 1[$, $\lambda \geq 0$ et pour $j \in \{1, ..., n\}$ et $i \in \{1, ..., m\}$, les coefficients p_j, c_i et $w_{i,j}$ sont tirés de manière uniforme dans, respectivement, $[0, U_p]$, $[0, U_c]$ et $[0, U_w]$. α, U_p, U_c et U_w sont des paramètres fixés par l'utilisateur.

Plus le paramètre λ est grand, plus le problème en inégalité est ardu car la partie purement liée au critère est écrasée par la satisfaction des contraintes. Le problème ainsi généré est donc d'autant plus difficile à résoudre que λ est grand.

4 Instances engendrées à partir de l'analyse de la transformée en $\mathbb{Z}$

Nous considérons dans cette section une autre approche qui permet d'engendrer des instances difficiles en programmation entière.

Principe On considère le problème général en nombre entier suivant :

$$\mathbb{P}_d : f_d(c,p) = \max\{p^t.x \mid Wx = c, x \in \mathbb{N}^n\}, \tag{7}$$

ainsi que le problème associé suivant (voir Lasserre [11]) :

$$\begin{aligned} \mathbb{I}_d : \hat{f}_d(c,p) &= \sum_{x \in \Omega(c)} e^{p^t.x}, \\ \Omega(c) &= \{x \in \mathbb{N}^n \mid W.x = c\}. \end{aligned} \tag{8}$$

Une partie des travaux de Lasserre ([10], [11] et [12]) suggère de chercher un dual de $\mathbb{P}_d$ à partir de la transformé en $\mathbb{Z}$ de $\mathbb{I}_d$. Sachant que $\mathbb{P}_d$ et $\mathbb{I}_d$ sont liés par la relation suivante :

$$f_d(c,p) = \lim_{r\to\infty} \frac{1}{r}.ln\left(\hat{f}_d(c,r.p)\right). \tag{9}$$

La transformée en $\mathbb{Z}$ de $\mathbb{I}_d$, par sa transformée inverse, mène à une nouvelle formulation équivalente du problème $\mathbb{I}_d$. A partir de ce dual de $\mathbb{I}_d$, l'utilisation de l'équation (9) permet de construire un dual pour le problème $\mathbb{P}_d$. La transformée en $\mathbb{Z}$ de $\mathbb{I}_d$ nous donne donc (voir Lasserre [10]) :

$$\begin{gathered} \hat{F}_d(z,p) = \sum_{y\in\mathbb{Z}^m} z^{-y}.\hat{f}_d(y,p) = \prod_{k=1}^{n} \frac{1}{(1-e^{p_k}.z^{-W^k})}, \\ avec\ pour\ k \in \{1,...,n\},\ z^{-W^k} = z_1^{-w_{1,k}}.z_2^{-w_{2,k}}...z_m^{-w_{m,k}}. \end{gathered} \tag{10}$$

On constate que $\hat{F}_d(z,p)$ est bien définie si $pour\ k \in \{1,...,n\},\ |z^{-W^k}| > e^{p_k}$. L'intégration de cette fonction (transformée en $\mathbb{Z}$ inverse) sera d'autant plus difficile qu'elle a de pôles. L'idée est donc d'engendrer un problème $\mathbb{P}_d$ conduisant à un grand nombre de pôles.

On considère $\sigma = \{j_1,...,j_m\} \subset \{1,...,m\}$ tel que la matrice $W_\sigma = (W^{j_1},...,W^{j_m})$ soit une base du problème linéaire associé. Pour un tel σ, nous avons le système à m équations suivant :

$$Pour\ j \in \sigma,\ z_1^{-w_{1,j}}.z_2^{-w_{2,j}}...z_m^{-w_{m,j}} = e^{p_j}. \tag{11}$$

Ce système a $\rho(\sigma) = det(W_\sigma)$ solutions de la forme (voir Lasserre [10]) :

$$\begin{gathered} z(k) = e^{\lambda}.e^{2\imath\pi\theta(k)},\ k \in \{1,...,\rho(\sigma)\} \\ avec\ \theta(k) \in \mathbb{R}^m\ et\ \lambda = W_\sigma^{-1}.p_\sigma. \end{gathered} \tag{12}$$

Donc, pour avoir un grand nombre de pôles, on doit engendrer des matrices de base W_σ ayant un déterminant suffisamment grand. Plus les rayons, $\lambda = W_\sigma^{-1}.p_\sigma$, des multi-disques de $\mathbb{C}^n$, sur lesquels sont ces pôles, sont proches les uns des autres plus l'intégration est difficile.

Dans le cas des problèmes uni-contraints, le calcul de l'ensemble des λ, pour toutes les bases admissibles, nous donne :

$$\lambda_k = \frac{p_k}{w_k}\ avec\ k \in \left\{ j \in \{1,...,n\} \mid w_j \neq 0\ et\ \frac{p_j}{w_j} \geq 0 \right\}. \tag{13}$$

Et on constate que plus les rapports définis par λ_k sont proches les uns des autres, plus le problème à une contrainte est difficile à résoudre. Cette conclusion peut-être généralisée au cas multicontraint grâce à l'utilisation de la transformée en $\mathbb{Z}$ en prenant les $W_\sigma^{-1}.p_\sigma$ proches :

$$\left\| W_\sigma^{-1}.p_\sigma - W_{\sigma'}^{-1}.p_{\sigma'} \right\| \leq \epsilon,\ \epsilon\ \in \mathbb{R}_+. \tag{14}$$

A partir du problème en égalité ainsi construit, on remplace simplement les égalités par des inégalités pour obtenir le problème inégalité désiré.

Application au cas à deux contraintes Pour étudier le temps de résolution d'un problème généré de cette façon, nous avons appliqué cette méthode au cas des problèmes à deux contraintes. Afin de simplifier les calculs, ces problèmes seront générés sous la forme :

$$(MKP)\begin{cases} max\ p_0.x_1 + (p_0+\delta_p).x_2 + ... + (p_0+(n-1).\delta_p).x_n, \\ s.c. \begin{cases} a_0.x_1 + (a_0+\delta_a).x_2 + ... + (a_0+(n-1).\delta_a).x_n \leq a, \\ b_0.x_1 + (b_0+\delta_b).x_2 + ... + (b_0+(n-1).\delta_b).x_n \leq b, \\ x_j \in \{0,1\},\ j \in \{1,...,n\}, \end{cases} \end{cases} \tag{15}$$

avec $n >= 3$.

Avec $\{j1, j2\} \subset \{0, ..., n-1\}$, considérons la matrice carré :

$$W_{j1,j2} = \begin{pmatrix} a_0 + j1.\delta_a & a_0 + j2.\delta_a \\ b_0 + j1.\delta_b & b_0 + j2.\delta_b \end{pmatrix} \tag{16}$$

$W_{j1,j2}$ est une base de $\mathbb{R}^2$ si :

$$|det(W_{j1,j2})| = |j1 - j2|.|\delta_a.b_0 - \delta_b.a_0| \neq 0, \tag{17}$$

or $|det(W_{j1,j2})| \geq |\delta_a.b_0 - \delta_b.a_0|$, il suffit alors de prendre $|\delta_a.b_0 - \delta_b.a_0| = DET \neq 0$. Toute les matrices $W_{j1,j2}$ sont ainsi des matrices de bases de $\mathbb{R}^2$.

Le respect de l'équation (14) nous donne la condition suivante :

$$\left\| (W_{j1,j2})^{-1}.\begin{pmatrix} p_o + j1.\delta_p \\ p_o + j2.\delta_p \end{pmatrix} - (W_{j3,j4})^{-1}.\begin{pmatrix} p_o + j3.\delta_p \\ p_o + j4.\delta_p \end{pmatrix} \right\| \leq \epsilon, \tag{18}$$

avec $\{j1, j2\} \subset \{0, ..., n-1\}$, $\{j3, j4\} \subset \{0, ..., n-1\}$ et $\{j1, j2\} \neq \{j3, j4\}$.

Afin de simplifier les calculs, cette condition sera imposée pour $j1 = j \in \{0, ..., n-3\}$, $j2 = j+1$, $j3 = j2 = j+1$ et $j4 = j3+1 = j+2$, ce qui donne :

$$\left\| (W_{j,j+1})^{-1}.\begin{pmatrix} p_o + j.\delta_p \\ p_o + (j+1).\delta_p \end{pmatrix} - (W_{j+1,j+2})^{-1}.\begin{pmatrix} p_o + (j+1).\delta_p \\ c_o + (j+2).\delta_p \end{pmatrix} \right\| \leq \epsilon, \tag{19}$$

ce qui donne :

$$\alpha.\delta_p^2 + 2.\beta.\delta_p + \gamma \leq (r.DET)^2, \tag{20}$$

avec

- $\alpha = 2.(a_0 - b_0 + \delta_a)^2 + (\delta_b - \delta_a)^2(8.j^2 + 12.j + 5) - 2.(\delta_b - \delta_a).(a_0 - b_0 + \delta_a).(4.j + 3)$,
- $\beta = -p_0.(\delta_b - \delta_a).(2.(a_0 - b_0 + \delta_a) - (\delta_b - \delta_a).(4.j + 3))$ et
- $\gamma = 2.p_0^2.(\delta_b - \delta_a)^2$.

En se plaçant sous les conditions suivantes :

- $\delta_b - \delta_a \leq 0$ et
- $2.(a_0 - b_0 + \delta_a) - (\delta_b - \delta_a).(4.(n-3) + 3) \leq 0$,

on a les majorations suivantes :

- $\alpha \leq \alpha' = 2.(a_0 - b_0 + \delta_a)^2 + (\delta_b - \delta_a)^2(8.(n-3)^2 + 12.(n-3) + 5)$ et
- $\beta \leq \beta' = -p_0.(\delta_b - \delta_a).(2.(a_0 - b_0 + \delta_a) - (\delta_b - \delta_a).(4.(n-3) + 3))$.

Ainsi en prenant $\alpha'.\delta_p^2 + 2.\beta'.\delta_p + \gamma \leq (r.DET)^2$, l'inéquation (20) est vérifiée quelque soit $j \in \{0, ..., n-3\}$. De plus la fonction $f:\ \delta_p \to \alpha'.\delta_p^2 + 2.\beta'.\delta_p + \gamma$ est minimale pour $\delta_p^* = -\frac{\beta'}{\alpha'}$. Nous devons donc avoir $(r.DET)^2 \geq f(\delta_p^*)$. On se placera alors dans le cas extrême où $\delta_p = \delta_p^*$, et les conditions prises précédemment nous garantissent $\delta_p^* \geq 0$.

De manière synthétique :

- δ_a et δ_b sont tirés aléatoirement dans $[1, U_\delta]$ tel que $\delta_b - \delta_a \leq 0$,
- p_0 est tiré aléatoirement dans $[1, U_p]$,
- a_0 et b_0 sont construits tels que

$$\begin{cases} \delta_a.b_0 - \delta_b.a_0 = DET \geq 0\ et \\ 2.(a_0 - b_0 + \delta_a) - (\delta_b - \delta_a).(4.(n-3) + 3) \leq 0, \end{cases}$$

- $\delta_p = -\frac{\beta'}{\alpha'}$ et
- les capacités a et b sont construites de la même façon que dans le cas des problèmes non corrélés.

5 Tests Numériques

L'ensemble des tests préliminaires présentés ici ont été menés sur une Sun Blade 100 sous système Solaris (500 MHz). Les instances engendrées par les techniques décrites dans les sections 3 et 4 ont été résolues de manière exacte avec CPLEX 9.0. Afin d'apprécier la difficulté de résolution de ces instances nous les avons comparées avec les instances présentées à la section 2.

Nous avons engendré des instances non-corrélées, I_NC, et fortement corrélées, I_FC, de la manière suivante :
- le nombre de variable $n \in \{50, 100, 150, 200, 250, 300\}$,
- pour chaque n, 25 problèmes ont été générés,
- le nombre de contraintes $m = 5$,
- les coefficients sont tirés aléatoirement dans $I = [1, 10]$ et dans $I = [1, 1000]$,
- $\epsilon = 0,5$ et
- $sc = 5$ (coefficient de corrélation).

Nous considérons des problèmes de taille raisonnable afin de conserver des temps de résolution exploitables mais qui permettent néanmoins de mettre en évidence la pertinence des techniques que nous avons développées afin d'engendrer des instances difficiles.

5.1 Instances engendrées à partir d'un problème combinatoire en contrainte égalité

Afin de montrer l'intérêt de cette approche, nous avons engendré, selon le principe énoncé dans la section 3, des instances non-corrélées et fortement corrélées. Nous appellerons ces instances respectivement I_L_NC(λ) et I_L_FC(λ), avec λ le multiplicateur de Lagrange. Lorsque le temps de calcul excède une heure, nous arrêtons la résolution.

Les tests sont présentés dans les tableaux 1 et 2. On constate que même pour les multiplicateurs de Lagrange assez faibles (et non nuls), nos instances nécessitent des temps de résolution plus importants que pour le cas $\lambda = 0$. Plus ce coefficient sera pris grand, plus on se rapproche du problème en contrainte égalité et donc plus les instances sont difficiles.

n	I_L_NC(0)	I_L_NC(5)	I_L_NC(10)	I_L_FC(0)	I_L_FC(5)	I_L_FC(10)
50	0,06	4,63	4,93	0,19	5,21	7,85
100	0,06	9,39	11,65	2,66	6,39	10,91
150	0,08	13,53	14,24	3,82	9,76	18,17
200	0,10	15,59	17,98	5,70	12,09	28,89
250	0,11	18,89	20,34	8,63	23,27	36,47
300	0,17	20,13	25,41	14,83	26,41	41,52

Tab. 1. Temps de calcul (en s.) pour $I = [1, 10]$.

n	I_L_NC(0)	I_L_NC(1)	I_L_NC(5)	I_L_FC(0)	I_L_FC(1)
50	0,11	8,23	2641,65	101,65	>1h
100	0,87	288,21	>1h	2031,14	>1h
150	2,26	1069,99	>1h	>1h	>1h
200	4,58	3586,76	>1h	>1h	>1h
250	10,56	>1h	>1h	>1h	>1h
300	10,91	>1h	>1h	>1h	>1h

Tab. 2. Temps de calcul (en s.) pour $I = [1, 1000]$.

5.2 Instances engendrées à partir de l'analyse de la transformée en $\mathbb{Z}$

Dans cette partie, nous mettons en avant l'intérêt de cette approche. Nous nous sommes limités à 2 contraintes. Les coefficients δ_a, δ_b et p_0 ont été tirés aléatoirement dans l'intervalle $[1, 10]$. Ces problèmes demandant des temps de résolution plus importants, nous avons considéré des instances pour lesquelles $n \in \{50, 100, 150, 200\}$; nous nommerons ces instances I_Z.

Les temps de résolution ont été comparés à des instances de problèmes fortement corrélés avec $I = [1, 1000]$ et $m = 2$. Les résultats sont présentés dans le tableau 3. Nous arrêtons la résolution lorsque le temps de calcul dépasse une heure.

Les résultats sont présentés dans le tableau 3. On constate qu'à taille égale nous obtenons des temps de résolution beaucoup plus importants pour des problèmes engendrés par des techniques faisant appel à la transformée en $\mathbb{Z}$.

n	I_FC	I_Z
50	6,19	8,73
100	9,66	426,46
150	25,82	>1h
200	175,22	>1h

Tab. 3. I_Z : Temps de calcul (en s.).

6 Conclusion

Les instances que nous avons présentées doivent nous permettre d'engendrer par deux techniques différentes des problèmes difficiles.

Les problèmes dérivées d'un problème combinatoire en contrainte égalité permettent de rendre plus difficile les problèmes à données non-corrélées et fortement corrélées. Même pour des λ faibles, les temps de résolution obtenues sont plus importants en comparaison avec des problèmes de taille identique. Il en est de même en ce qui concerne les problèmes engendrés à partir de l'analyse de la transformée en $\mathbb{Z}$ lorsque les coefficients sont petits.

Les résultats préliminaires obtenus sont assez prometteurs et montrent l'intérêt de ces approches. Les deux interprétations faites sur les ratios profit sur poids proches, dans le cas à une contrainte et sur les matrices de base dans le cas multicontraint avec l'emploi de la transformée en $\mathbb{Z}$, semblent engendrer des instances plus ardues à résoudre. L'approche de la transformée en $\mathbb{Z}$ appliquée à deux contraintes peut-être généralisée aux problèmes à plus de deux contraintes.

Références

1. Boyer, V., El Baz, D. et Elkihel, M. : A new heuristic for the multidimensional knapsack problem. 7ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'06), Lille (France) (2006).
2. Boyer, V., El Baz, D. et Elkihel, M. : An exact method for the 0-1 multidimensional knapsack problem. 8ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'06), Grenoble (France) (2007).
3. Chu, P. et Beasley, J. : Genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics, vol. 4, pages 63-86 (1998).
4. El Baz, D., Elkihel, M. et Gely, L. : Résolution efficace de problèmes en variables 0-1 avec contraintes en égalité. ROADEF'05, pages 15-27 (2005).
5. Freville, A. : The multidimensional 0-1 knapsack problem : An overview. European Journal of Operational Research, vol. 155, pages 1-21 (2004).
6. Fisher, M. : The Lagrangean Relaxation Method for Solving Integer Programming Problems. Manadgement Science, vol. 27, pages 1-18 (1981).

7. Geoffrion, A. et Gomory, E. : The theory and computation of knapsack functions. Operations Research, vol. 17, pages 437-454 (1966).
8. Geoffrion, A. et Gomory, E. : The Lagrangean Relaxation for Integer Programming. Mathematical Programming Study, vol. 2, pages 82-114 (1974).
9. Kellerer, H., Pferschy, U. &Pisinger, D. : Knapsack Problems. Springer (2004).
10. Lasserre, J. : Generating functions and duality for integer programs. Discrete Optimization, vol. 1, pages 167-187 (2003).
11. Lasserre, J. : Optimization : Structure and Applications. Kluwer Academic Publishers (2003).
12. Lasserre, J. : Integer programming duality. Proceedings of Symposia in Applied Mathematics, vol. 61, pages 67-76 (2004).
13. Martello,S. et Toth, P. : Upper bounds and algorithms for hard 0-1 knapsack problems. Operations Research, vol. 45, pages 768-778 (1997).
14. Nemhauser, L. et Wolsey, A. : Integer and combinational optimization. Wiley Interscience (1988).
15. Pandit, S. Ravi Kumar, M. : A lexicographic search for strongly correlated 0-1 knapsack problems. Operations Research, vol. 30, pages 97-116 (1993).
16. Pisinger, D. : A minimal algorithm for the Bounded Knapsack Problem. Proceedings of Integer Programming and Combinatorial Optimization, Fourth IPCO conference, Balas, J. & Clausen, E., editor, vol. 920 of Lecture Notes in Computer Science, Springer, Berlin, pages 95-109 (1995).