Une résolution efficace d'une reformulation linéaire de programmes quadratiques en nombres entiers

Alain Billionnet[1], Sourour Elloumi[2], Amélie Lambert[2]

1.CEDRIC-ENSIIE, 18 allée Jean Rostand, F-91025 Evry cedex, France
2. CEDRIC-CNAM, 292 rue Saint-Martin, F-75141 Paris cedex 03, France

1 Présentation du problème

De nombreux problèmes de recherche opérationnelle peuvent se modéliser sous la forme d'un programme mathématique à valeurs entières dont la fonction objectif est quadratique et est soumise à des contraintes linéaires. Un problème de ce type peut s'écrire sous la forme :

$$(QP)\begin{cases} Min\ f(x) = x^TQx + c^Tx \\ s.c \quad x \in X \subset \mathbb{N}^n \end{cases}$$

Avec $X = \{Ax = b\,(1),\ Dx \leq e\,(2),\ 0 \leq x_i \leq u_i\,(3),\ x_i \in \mathbb{N}\,(4),\ i = 1, \ldots, n\}$.
Ce problème appartient à la classe des problèmes NP-difficiles [1].

2 Une reformulaion linéaire : `BIL (Binary Integer Reformulation)`

Une façon de reformuler (QP) en un programme dont la fonction objectif est linéaire est de lui appliquer l'appoche `BIL` [2]. L'idée est d'abord de décomposer chacunes des variables entières x_i en puissances de 2. Puis dans chaque produit x_ix_j de remplacer la variable x_i par sa décomposition binaire. Chaque produit x_ix_j devient donc une expression de produits de variables `entière` par `binaire` que nous linéarisons. Soit $(LP_{\texttt{BIL}})$ le problème suivant :

$$(LP_{\texttt{BIL}})\begin{cases} Min\ f(x,y) = \sum\limits_{i=1}^{n} \sum\limits_{\substack{j=1 \\ q_{ij}\neq 0}}^{n} q_{ij}y_{ij} + \sum\limits_{i=1}^{n} c_ix_i \\ s.c. \quad (1),(2) \\ \qquad\ \ x,y,z,t \in P_{xyzt} \end{cases}$$

Avec P_{xyzt}

$$P_{xyzt}\begin{cases} & (2) & \\ & x_i = \sum\limits_{k=0}^{\lfloor log(u_i)\rfloor} 2^k t_{ik} & i \in I & (5) \\ & (5),(13) & \\ & z_{ijk} \leq u_jt_{ik} & (i,k) \in E,\ j \in I,\ q_{ij} < 0 & (6) \\ & z_{ijk} \leq x_j & (i,k) \in E,\ j \in I,\ q_{ij} < 0 & (7) \\ & z_{ijk} \geq x_j - u_j(1 - t_{ik}) & (i,k) \in E,\ j \in I,\ q_{ij} > 0 & (8) \\ x,y,z,t: & z_{ijk} \geq 0 & (i,k) \in E,\ j \in I,\ q_{ij} > 0 & (9) \\ & y_{ij} = \sum\limits_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{ijk} & i,j \in I,\ q_{ij} \neq 0 & (10) \\ & \sum\limits_{k=0}^{\lfloor log(u_i)\rfloor} 2^k z_{ijk} = \sum\limits_{l=0}^{\lfloor log(u_j)\rfloor} 2^l z_{jil} & i,j \in I,\ q_{ij} \neq 0 & (11) \\ & y_{ij} = u_ix_j + u_jx_i - u_iu_j & i,j \in I,\ q_{ij} \neq 0 & (12) \\ & t_{ik} \in \{0,1\} & (i,k) \in E & (13) \end{cases}$$

$(LP_{\texttt{BIL}})$ est un programme linéaire mixte entier qui est équivalent à (QP), où chaque variable $y_{ij} = x_ix_j$, et chaque variable $z_{ijk} = x_jt_{ik}$. $(LP_{\texttt{BIL}})$ est une programme de grande taille, en notant N le nombre de variable t, sa taille est en $O(Nn)$.

Soit $\overline{P_{xyzt}}$ le polyhèdre obtenu en relâchant les contraintes d'intégrités de P_{xyzt}, i.e. replacer les contraintes (13) par $0 \leq t_{ik} \leq 1$. Nous noterons $(\overline{LP_{\texttt{BIL}}})$ la relaxation continue de $(LP_{\texttt{BIL}})$.

Proposition 21 *La projection de $\overline{P_{xyzt}}$ sur x et y est le polyhèdre suivant P_{xy} :*

$$P_{xy}\begin{cases} & (3) & \\ & y_{ij} = y_{ji} & i,j \in I,\ q_{ij} \neq 0\ (14) \\ & y_{ij} \leq u_jx_i & i,j \in I,\ q_{ij} \neq 0\ (15) \\ x,y: & y_{ij} \leq u_ix_j & i,j \in I,\ q_{ij} \neq 0\ (16) \\ & y_{ij} \geq x_iu_j + x_ju_i - u_iu_j & i,j \in I,\ q_{ij} \neq 0\ (17) \\ & y_{ij} \geq 0 & i,j \in I,\ q_{ij} \neq 0\ (18) \end{cases}$$

Idée de preuve : *Il est évident que* $\forall (x,y,z,t) \in \overline{LP_{\texttt{BIL}}}$, $(x,y) \in P_{xy}$. *Réciproquement, si* $(x,y) \in P_{xy}$, *soit* $\bar{u}_i = \sum_{k=0}^{\lfloor log(u_i) \rfloor} 2^k$ *,prenons* $t_{ik} = x_i/\bar{u}_i$, *et* $z_{ijk} = t_{ik}x_j$, *on montre facilement que* $(x,y,z,t) \in \overline{LP_{\texttt{BIL}}}$.

$$\text{Soit } (P) : \ (P) \begin{cases} Min\ f(x,y) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ q_{ij} \neq 0}}^{n} q_{ij}y_{ij} + \sum_{i=1}^{n} c_i x_i \\ s.c. \quad (1), (2) \\ \qquad x, y \in P_{x,y} \end{cases}$$

le programmme (P) à donc la même valeur optimale continue que $(\overline{LP_{\texttt{BIL}}})$. Seulement, la taille de (P) qui est en $O(n^2)$ est significativement plus petite que celle de $(\overline{LP_{\texttt{BIL}}})$

3 Un nouvel algorithme de branch and bound

Dans cette partie nous proposons un algorithme de Branch and Bound qui permet de trouver la solution entière de $(LP_{\texttt{BIL}})$ en résolvant à chaque noeud de l'arbre le problème (P). Nous pouvons en effet remarquer qu'une solution admissible de $(LP_{\texttt{BIL}})$ est une solution entière de (P), qui a la propriété suivante $\forall\ i,j,\ x_i x_j = y_{ij}$. L'idée est donc de parcourir l'arbre de résolution, et de résoudre à chaque noeud le problème (P) en nombre entiers, si tous les produits $x_i x_j$ sont égaux aux variables y_{ij}, on a une solution admissible de $(LP_{\texttt{BIL}})$, et donc une borne supérieure de $f(x,y)$. Par contre si $\exists\ i,j$ tel que le produit $x_i x_j = v_i v_j$ n'est pas égal à la variable $y_{ij} = v_{ij}$, on branche de la façon suivante :

1. **Branche 1 :** On ajoute les contraintes : $x_i = v_i$, $x_j = v_j$ et $y_{ij} = v_i v_j$.
2. **Branche 2 :** On ajoute les contraintes : $x_i = v_i$ et $x_j \neq v_j$.
3. **Branche 3 :** On ajoute les contraintes : $x_i \neq v_i$ et $x_j = v_j$.

Cet algorithme est efficace sur des problèmes de grande taille. En effet, si (QP) possède moins de 30 variables entières, il est préférable de résoudre $(LP_{\texttt{BIL}})$ par un solveur standard. Par contre, plus la taille du problème de départ augmente plus il est intéressant, en terme de temps de calcul d'utiliser cet algorithme de branch and bound comme méthode de résolution.

Références

1. M.R. Garey and D.S. Johnson, *Computers and Intractability : A guide to the theory of NP-Completness*, Freeman, (1979).
2. A. Billionnet, S. Elloumi et A. Lambert. *Linear Reformulations of Integer Quadratic Programs*, In MCO 2008, september [8-10], 43-51, (2008)
3. R. Fortet *Applications de l'Algèbre de Boole en Recherche Opérationelle*, Revue Française De Recherche Opérationelle [4] 17-25, (1960)