

# Reformulation convexe des programmes quadratiques entiers : un algorithme de Branch and Bound fondé sur la structure du problème reformulé

Alain Billionnet<sup>1</sup>, Sourour Elloumi<sup>1</sup>, Amélie Lambert<sup>1</sup>

CEDRIC-ENSIIE, 1 square de la résistance, F-91025 Evry cedex, France  
 {alain.billionnet,sourour.elloumi,amelie.lambert}@ensiie.fr

**Mots-Clés :** *programmation quadratique entière, reformulation convexe, Branch and Bound*

## Une reformulation quadratique convexe : IQCR (Integer Quadratic Convex Reformulation)

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 :

$$(QP) \begin{cases} \text{Min} & f(x) = x^T Q x + c^T x \\ \text{s.c.} & Ax = b \quad Dx \leq e \quad 0 \leq x \leq u \quad x \in \mathbb{N}^n \end{cases}$$

La reformulation IQCR [1] consiste en la définition d'un schéma de reformulations convexes de (QP), puis au calcul au sein de ce schéma d'une reformulation convexe optimale du point de vue de la borne obtenue par relaxation continue. Cette reformulation optimale est calculée par les solutions duales d'un relaxation semi-définie de (QP). Concrètement, nous définissons la fonction suivante :

$$f_{\alpha,\beta}(x, y) = f(x) + \sum_{i=1}^n \sum_{j=1}^n \beta_{ij} (x_i x_j - y_{ij}) + \alpha \sum_{r=1}^m \left( \sum_{i=1}^n a_{ri} x_i - b_r \right)^2$$

Il est clair que  $f_{\alpha,\beta}(x, y) = f(x)$  si  $y_{ij} = x_i x_j$ . Nous forçons cette égalité par la linéarisation utilisée dans [2] qui consiste à exprimer chaque produit  $x_i x_j$  en une combinaison de termes quadratiques dont chacun est le produit d'une variable entière par une variable binaire. Puis, nous linéarisons ces termes quadratiques. Dès lors que  $f_{\alpha,\beta}(x, y)$  est convexe, nous obtenons ainsi un programme quadratique convexe en nombres entiers noté  $(QP_{\alpha,\beta})$ .

Cette reformulation ajoute un nombre important de contraintes et de variables. Cependant, la relaxation continue de  $(QP_{\alpha,\beta})$  possède une propriété de projection qui permet d'en réduire la taille. En effet, en projetant sur les variables  $x$  et  $y$  le polyèdre de  $(QP_{\alpha,\beta})$ , obtenu en relâchant ses contraintes d'intégrité, nous obtenons le programme  $(P_{\alpha,\beta})$  suivant :

$$(P_{\alpha,\beta}) \begin{cases} \text{Min} & f_{\alpha,\beta}(x, y) \\ \text{s.c.} & Ax = b \\ & Dx \leq e \\ & y_{ij} = y_{ji} & \forall i, \forall j \\ & y_{ij} \leq u_j x_i & \forall i, \forall j \\ & y_{ij} \leq u_i x_j & \forall i, \forall j \\ & y_{ij} \geq x_i u_j + x_j u_i - u_i u_j & \forall i, \forall j \\ & y_{ij} \geq 0 & \forall i, \forall j \end{cases}$$

La valeur optimale de  $(P_{\alpha,\beta})$  est égale à la valeur optimale de la relaxation continue de  $(QP_{\alpha,\beta})$ . Seulement, la taille de  $(P_{\alpha,\beta})$ , qui est en  $O(n^2)$ , est significativement plus petite que celle de  $(QP_{\alpha,\beta})$  [1].

### Un algorithme spécifique de Branch and Bound

Nous proposons un algorithme de Branch and Bound, basé sur cette propriété de projection, qui permet de trouver la solution optimale de  $(QP)$ , i.e. de  $(QP_{\alpha,\beta})$ , en résolvant à chaque noeud de l'arbre un problème de plus petite taille. En notant  $(P_{\alpha,\beta}^e)$ , le problème  $(P_{\alpha,\beta})$  avec la contrainte  $x \in \mathbb{N}^n$ , nous avons la propriété suivante :

**Proposition 1** *Une solution optimale de  $(QP)$ , i.e. de  $(QP_{\alpha,\beta})$ , peut être trouvée en résolvant le programme  $(P_{\alpha,\beta}^e)$  augmenté des contraintes  $x_i x_j = y_{ij}$*

Ainsi une manière de trouver une solution optimale de  $(QP)$  est de résoudre à chaque noeud de l'arbre de résolution le problème  $(P_{\alpha,\beta}^e)$ , puis de brancher de la manière suivante :

Soit  $(\bar{x}, \bar{y})$  une solution du noeud courant, plusieurs cas de figure se présentent :

1. Si  $(\bar{x}, \bar{y})$  respecte la propriété  $\forall i, \forall j, \bar{x}_i \bar{x}_j = \bar{y}_{ij}$ , alors  $\bar{x}$  est une solution réalisable de  $(QP)$  et  $f(\bar{x}) = f_{\alpha,\beta}(\bar{x}, \bar{y})$
2. Si  $\exists i, j$  tels que  $\bar{x}_i \bar{x}_j \neq \bar{y}_{ij}$ , nous branchons de la façon suivante :
  - a. La branche où  $x_i \neq \bar{x}_i$ . Cette branche donne naissance à deux sous arbres, dans le premier on ajoute à la formulation  $(P_{\alpha,\beta}^e)$  la contrainte  $x_i \leq \bar{x}_i - 1$  et dans le deuxième la contrainte  $x_i \geq \bar{x}_i + 1$ .
  - b. La branche où  $x_i = \bar{x}_i$  et  $x_j \neq \bar{x}_j$ . Cette branche donne naissance à deux sous arbres, dans le premier on ajoute à la formulation  $(P_{\alpha,\beta}^e)$  les contraintes  $x_i = \bar{x}_i$  et  $x_j \leq \bar{x}_j - 1$  et dans le deuxième les contraintes  $x_i = \bar{x}_i$  et  $x_j \geq \bar{x}_j + 1$ .
  - c. La branche où  $x_i = \bar{x}_i$ ,  $x_j = \bar{x}_j$  et  $y_{ij} = \bar{x}_i \bar{x}_j$ . Dans le sous arbre, on ajoute à la formulation  $(P_{\alpha,\beta}^e)$  les contraintes  $x_i = \bar{x}_i$ ,  $x_j = \bar{x}_j$  et  $y_{ij} = \bar{x}_i \bar{x}_j$ .

### Expérimentations

Nous avons testé cet algorithme sur des instances de programmes quadratiques soumis à une contrainte d'égalité qui ont été générées aléatoirement et déjà utilisées dans [1] et [2]. Malgré une implémentation naïve, cet algorithme s'avère compétitif avec la résolution directe du problème non projeté  $(QP_{\alpha,\beta})$ , par un solveur standard MIQP. Nos résultats expérimentaux montrent qu'il a un temps moyen de résolution divisé par un facteur 2 par rapport à la résolution de  $(QP_{\alpha,\beta})$  par un solveur MIQP.

### Références

- [1] Billionnet, A., Elloumi, S., Lambert, A. : Solution of Integer Quadratic Programs through convex reformulation. Mathematical Programming. Article en révision.
- [2] Billionnet, A., Elloumi, S., Lambert, A. : Linear Reformulations of Integer Quadratic Programs. MCO 2008, september 8-10, 43-51 (2008)