# Optimisation de programmes polynomiaux en variables 0-1 et sans contraintes

Claudia d'Ambrosio, Sourour Elloumi, Amélie Lambert, Arnaud Lazare

# Optimisation de programmes polynomiaux en variables 0-1 et sans contraintes

Claudia D'Ambrosio[1], Sourour Elloumi[2,3], Amélie Lambert[3], Arnaud Lazare[2,3]

[1] LIX - 91128 Palaiseau CEDEX, France

[2] ENSTA ParisTech, 91120 Palaiseau, France

arnaud.lazare@ensta-paristech.fr

[3] CEDRIC-Cnam - Paris Cedex 03, France

**Mots-clés** : *Programmation polynomiale en variables binaires, Reformulation convexe*

## 1 Introduction

Dans ce papier nous nous intéressons à minimiser un polynôme de degré maximum 4 dont les variables sont binaires. La formulation d'un tel problème est la suivante :

$$(P)\begin{cases} \min f(x) = \displaystyle\sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i}} c_p x_i + \sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i x_j}} c_p x_i x_j + \sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i x_j x_k}} c_p x_i x_j x_k + \sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i x_j x_k x_l}} c_p x_i x_j x_k x_l \\ x \in \{0,1\}^n \end{cases}$$

Ce problème est $\mathcal{NP}$-difficile. Les difficultés proviennent ici à la fois de la non-convexité de la fonction objectif, mais aussi de l'intégrité des variables. Classiquement, un tel problème est résolu par un algorithme de branch-and-bound basé sur une relaxation de $(P)$ plus facile à résoudre [3, 1]. Ici, nous proposons deux méthodes de résolution basées sur des reformulations convexes de $(P)$. Dans la première approche, nous reformulons d'abord $(P)$ en un problème équivalent et quadratique, puis nous convexifions la fonction $f$ en utilisant la programmation semi-définie. Dans la deuxième méthode, nous convexifions directement la fonction $f$, c'est à dire en conservant son degré initial.

## 2 `Q_form Method` : Reformulation quadratique convexe de $(P)$

Ici, nous présentons une formulation équivalente à $(P)$ qui est quadratique convexe. Pour cela, nous ajoutons de nouvelles variables $y_{ij}$ égales aux produits $x_i x_j$, et nous linéarisons l'égalité $y_{ij} = x_i x_j$ par les inégalités de Fortet [4]. Ensuite, grâce aux variables $y$, nous remplaçons les monômes degrés 3 ou 4 par des monômes de degré 2. Plus précisément, chaque terme trilinéaire $x_i x_j x_k$ est remplacé par le terme quadratique $y_{ij} x_k$, et chaque terme quadrilinéaire $x_i x_j x_k x_l$ est remplacé par le terme quadratique $y_{ij} y_{kl}$. Nous obtenons ainsi la fonction a-priori non convexe suivante :

$$f'(x,y) = \sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i}} c_p x_i + \sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i x_j}} c_p x_i x_j + \sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i x_j x_k}} c_p y_{ij} x_k + \sum_{\substack{\text{monômes p}\\ \text{de la forme}\\ x_i x_j x_k x_l}} c_p y_{ij} y_{kl}$$

La deuxième étape consiste à convexifier $f'$. Pour cela, nous introduisons la fonction suivante pour tout $\lambda \in \mathbb{R}^n$ et $\lambda' \in \mathbb{R}^{n\times n}$ :

$$f_{\lambda,\lambda'}(x,y) = f'(x,y) + \sum_i \lambda_i (x_i^2 - x_i) + \sum_{i,j} \lambda'_{ij} (y_{ij}^2 - y_{ij})$$

qui a la même valeur que $f'$ puisque $x$ et $y$ sont des variables binaires. De plus, il existe des valeurs de $\lambda$ et $\lambda'$ pour lesquelles $f_{\lambda,\lambda'}$ est une fonction convexe. En utilisant la programmation semi-definie positive [2], nous déterminons des vecteurs $\lambda$ et $\lambda'$ optimaux, dans le sens où ils maximisent la valeur de la relaxation continue du problème reformulé. Le problème équivalent et convexe est ensuite résolu par un branch-and-bound.

## 3 `P_form Method` : Reformulation polynomiale convexe de $(P)$

Ici, nous reformulons $(P)$ directement en un problème convexe sans ajouter de variable ou de contrainte. L'idée est de perturber le Hessien de $f$ en lui ajoutant des fonctions nulles sur le domaine de $(P)$. Pour cela, étant donné un vecteur $\gamma \in \mathbb{R}^n$, nous construisons $f_\gamma$ qui a la même valeur que $f$ puisque $x$ est une variable binaire :

$$f_\gamma(x) = f(x) + \sum_i \gamma_i(x_i^2 - x_i)$$

Comme dans la section précédente, nous cherchons un vecteur $\gamma$ tel que $f_\gamma(x)$ soit une fonction convexe. Cependant, comme $f$ a un degré supérieur à 2, son hessien $H_f(x)$ n'est pas constant. Ainsi, dans un premier temps, nous approximons $H_f(x)$. Pour cela, nous calculons les intervalles $[\underline{c_{ij}},\overline{c_{ij}}]$ qui contiennent les éléments de $H_f(x)$, i.e. tels que $\forall(i,j), h_{ij} \in [\underline{c_{ij}},\overline{c_{ij}}]$. Ensuite, nous calculons le vecteur $\gamma$ grâce au Théorème de Gerschgorin de la façon suivante :

$$\gamma_i = \max\left[0, \left(-\underline{c_{ii}} + \sum_{j\neq i} \max\left(\left|\underline{c_{ij}}\right|, |\overline{c_{ij}}|\right)\right)\right] \quad \forall i \in \{1,...,n\}$$

Le problème reformulé est ensuite résolu par un branch-and-bound.

## 4 Conclusions et perspectives

Nous avons présenté deux approches pour résoudre $(P)$ qui sont basées sur des reformulations convexes. La première utilise la littérature sur la programmation quadratique, mais la taille du problème reformulé est supérieure à celle de $(P)$. La deuxième reste dans l'espace initial des variables, mais la difficulté est que nous devons travailler avec une approximation du Hessien non constant de $f$. Nous présenterons une comparaison expérimentale de ces approches.

Dans un premier temps, nous nous sommes intéressés au cas où les variables sont binaires, nous travaillons à l'extension de ces premiers résultats au cas où les variables peuvent être des entiers naturels ou des réels.

## Références


[1] C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method, $\alpha$bb, for general twice-differentiable constrained NLPs - I. Theoretical advances. *Computers and Chemical Engineering*, 22(9) :1137–1158, 1998.

[2] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109 :55–68, 2007.

[3] C. Buchheim and C. D'Ambrosio. Monomial-wise optimal separable underestimators for mixed-integer polynomial optimization. *Journal of Global Optimization*, pages 1–28, 2016.

[4] R. Fortet. L'algèbre de boole et ses applications en recherche opérationnelle. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 4 :5–36, 1959.