



HAL
open science

Projective Cutting-Planes : un algorithme de plans coupants à base de projections et de points intérieurs

Daniel Cosmin Porumbel

► **To cite this version:**

Daniel Cosmin Porumbel. Projective Cutting-Planes : un algorithme de plans coupants à base de projections et de points intérieurs. ROADEF 2019, Feb 2019, Le Havre, France. hal-03956869

HAL Id: hal-03956869

<https://hal.science/hal-03956869>

Submitted on 25 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Projective Cutting-Planes : un algorithme de plans coupants à base de projections et de points intérieurs

Daniel Porumbel¹

CEDRIC-CNAM, 292 rue Saint Martin, 70003 Paris, daniel.porumbel@cnam.fr

Mots-clés : plans coupants, projections dans un polytope

1 Introduction

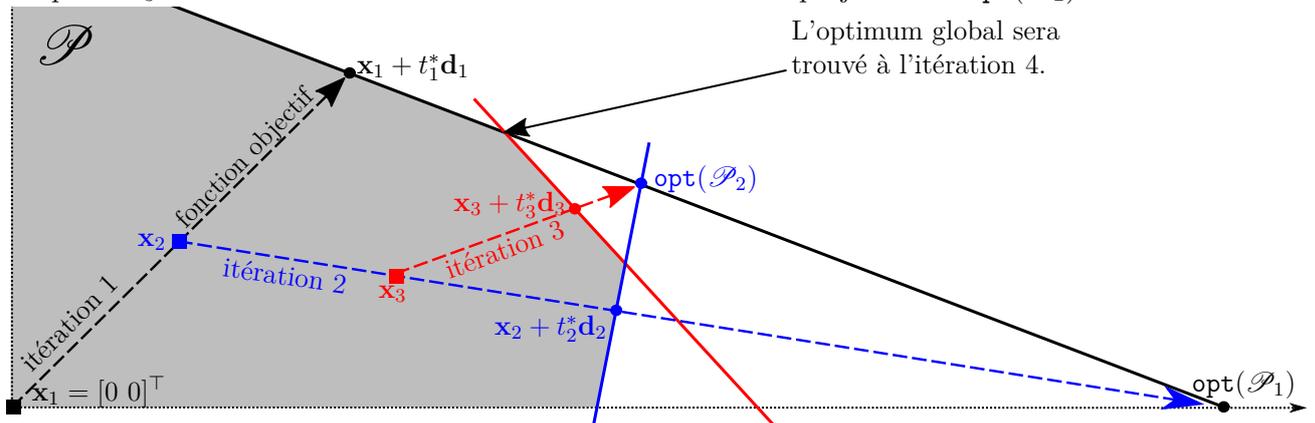
Soit un Programme Linéaire (PL) défini sur un polytope \mathcal{P} avec un nombre exorbitant (exponentiel) de contraintes.

$$\text{opt} \left\{ \mathbf{c}^\top \mathbf{x} : \mathbf{a}^\top \mathbf{x} \geq b, \forall (\mathbf{a}, b) \in \mathcal{C} \right\} = \text{opt} \left\{ \mathbf{c}^\top \mathbf{x} : \mathbf{x} \in \mathcal{P} \right\},$$

Pour résoudre ce type de PL, on fait souvent appel à un algorithme de plans coupants qui ajoute les contraintes une par une, à l'aide d'un sous-problème de séparation. Un des algorithmes de plans coupants les plus répandus considère à chaque itération un sous-ensemble de contraintes de \mathcal{C} et résout à l'optimalité le PL réduit associé, souvent à l'aide du **Simplexe**. À chaque itération it , ce sous-ensemble de contraintes détermine une enveloppe extérieure \mathcal{P}_{it} du polytope \mathcal{P} . Cette enveloppe est itérativement améliorée et on obtient $\mathcal{P}_1 \supseteq \mathcal{P}_2 \supseteq \dots \mathcal{P}$.

2 Les idées de base du Projective Cutting-Planes

La figure ci-dessus présente les trois premières itérations de l'algorithme proposé (**Projective Cutting-Planes**) sur un PL avec deux variables. À chaque itération it , cet algorithme projette un point intérieur $\mathbf{x}_{it} \in \mathcal{P}$ le long d'une direction \mathbf{d}_{it} jusqu'au point où on «heurte» une facette. À l'itération 1 sur cette figure, l'algorithme projette le point $\mathbf{x}_1 = \mathbf{0} = [0 \ 0]^\top$ le long de la fonction objectif $\mathbf{d}_1 = \mathbf{c}$, comme le montre la flèche noire en pointillés; cette première projection permet de découvrir aussi la contrainte noire. À l'itération $it = 2$, on prend le point \mathbf{x}_2 au milieu de cette flèche noire et on le projette vers la solution optimale $\text{opt}(\mathcal{P}_1)$ de l'enveloppe extérieure \mathcal{P}_1 — à l'itération 1, l'enveloppe extérieure $\mathcal{P}_1 \supseteq \mathcal{P}$ contient le plus grand triangle. Cette deuxième projection génère une deuxième facette (ligne continue bleue) qui s'ajoute aux facettes de \mathcal{P}_1 pour construire \mathcal{P}_2 . La troisième projection (en rouge) prend le point \mathbf{x}_3 au milieu entre le carré bleu et le cercle bleu et le projette vers $\text{opt}(\mathcal{P}_2)$.



À chaque itération, l'algorithme doit déterminer $t^* = \max \{t : \mathbf{x} + t\mathbf{d} \in \mathcal{P}\}$, c. à. d, il faut calculer une projection. Pour cela, il faut trouver une contrainte saturée (satisfaite avec égalité) par $\mathbf{x} + t^*\mathbf{d}$. Cela résout implicitement le sous-problème de séparation pour tous les points $\mathbf{x} + t\mathbf{d}$ avec $t \in \mathbb{R}_+$: les solutions $\mathbf{x} + t\mathbf{d}$ avec $t > t^*$ peuvent être séparées par la contrainte saturée mentionnée plus haut, alors que les points $\mathbf{x} + t\mathbf{d}$ avec $t \in [0, t^*]$ ne peuvent pas être séparés.

La plus grande difficulté réside dans le calcul efficace des projections. Comme la projection généralise la séparation, elle peut paraître beaucoup plus difficile à calculer. Ce n'est pas toujours le cas : j'ai proposé dans [1] quatre techniques (dans quatre contextes différents) qui permettent de calculer une projection presque aussi rapidement qu'une séparation.

1) Une première technique consiste à généraliser les idées principales de l'algorithme de séparation sans augmenter sa complexité. Nous avons pu appliquer cela pour un problème d'optimisation robuste avec un très grand nombre de coupes robustes, *i.e.*, pour chaque contrainte nominale $\mathbf{a}_{\text{nom}}^\top \mathbf{x} \geq b_{\text{nom}}$ il existe de nombreuses coupes robustes $\mathbf{a}^\top \mathbf{x} \geq b$ qui résultent d'une modification des coefficients nominaux, *i.e.*, \mathbf{a} est obtenu en modifiant au maximum Γ coefficient de \mathbf{a}_{nom} , où Γ est le budget d'incertitude. Si la séparation nécessite de parcourir toutes les contraintes nominales $\mathbf{a}_{\text{nom}}^\top \mathbf{x} \geq b_{\text{nom}}$ et minimiser une différence $\mathbf{a}^\top \mathbf{x} - b$, la projection demande de parcourir les mêmes contraintes pour minimiser un ratio $\frac{\mathbf{a}^\top \mathbf{x}}{b}$.

2) Une deuxième technique est applicable aux (nombreux) problèmes pour lesquels les contraintes de \mathcal{P} sont associées aux solutions réalisables d'un PL auxiliaire. C'est le cas des programmes issus de la décomposition de Benders, pour lesquels le sous-problème de séparation se réduit à un PL. Dans un tel contexte, la projection peut être formalisée sous la forme d'un programme Linéaire-Fractionnaire (PLF) : minimiser un ratio de deux fonctions linéaires soumises à des contraintes linéaires. En utilisant la transformation de Charnes-Cooper, il est possible de transformer ce PLF en un PL de même taille.

3) Une troisième technique est applicable aux (nombreux) problèmes pour lesquels les contraintes de \mathcal{P} sont associées aux solutions réalisables d'un PLNE. Cette situation concerne souvent les programmes duals issus des modèles à base de génération de colonnes, où on résout un PLNE pour trouver une nouvelle coupe duale (colonne primale). Pour calculer une projection, on utilise une transformation Charnes-Cooper discrétisée et on obtient un « Disjunctive LP ».

4) La dernière technique concerne les PLs où on peut trouver les contraintes de \mathcal{P} grâce à la programmation dynamique. On montre qu'on peut calculer une projection par programmation dynamique en utilisant un nombre similaire d'états que pour la séparation.

3 Conclusions et perspectives

La contrainte d'espace ne me permet pas de présenter de résultats numériques, mais je me permets de renvoyer le lecteur vers [1] pour des tests expérimentaux sur : (i) un problème d'optimisation robuste, (ii) un PL issu de la reformulation de Benders, (iii) un PL pour résoudre la coloration de graphe à l'aide de la génération de colonnes (iv) un modèle de génération de colonnes pour Multiple-Length Cutting-Stock. Le code source (13279 lignes) est disponible en ligne à cedric.cnam.fr/~porumbed/projcutplanes/.

Le principal avantage de l'algorithme proposé réside dans le fait qu'il génère une séquence de points intérieurs $\mathbf{x}_1, \mathbf{x}_2, \dots$ en plus des points extérieurs $\text{opt}(\mathcal{P}_1), \text{opt}(\mathcal{P}_2), \dots$; d'une certaine manière, il est plus similaire à un algorithme de point intérieur qu'à l'algorithme du Simplexe. Pour de nombreuses instances sur plusieurs problèmes, le nouvel algorithme nécessite 2 ou 3 fois moins d'itérations que la méthode classique. Il a trouvé une nouvelle borne inférieure pour une instance du problème (bien étudié) de coloration de graphe. Les techniques mentionnées ci-dessus pour calculer les projections pourraient être appliquées à d'autres problèmes.

Références

- [1] D. Porumbel. Cutting Planes by Projecting Interior Points onto Polytope Facets *Raport Technique CNAM*, papier soumis, voir <http://cedric.cnam.fr/~porumbed/main.pdf>