Semidefinite programming by projective-cutting-planes

Daniel Porumbel[1]

Conservatoire National des Arts et Métiers

daniel.porumbel@cnam.fr

Mots-clés : *optimisation semi-définie positive, plans coupants*

1 Introduction, synthèse et conclusion

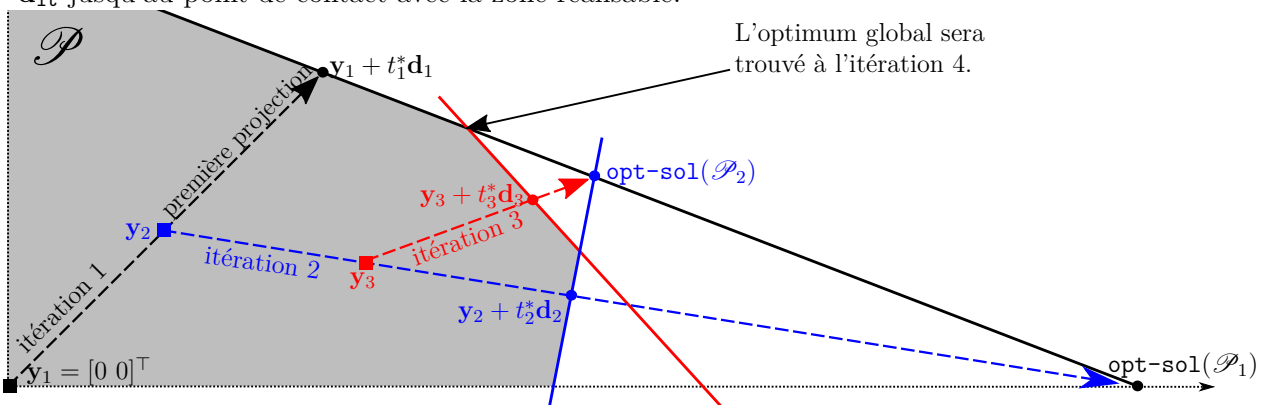
On considère le programme semi-défini positif (SDP) suivant :

$$\mathscr{P}\begin{cases}\max \ \mathbf{b}^\top \mathbf{y} & \text{(1a)}\\ s.t \ \ X = C - \mathcal{A}^\top \mathbf{y} & \text{(1b)}\\ \quad X \succeq \mathbf{0} \iff X \bullet \mathbf{s}\mathbf{s}^\top \geq 0 \ \forall \mathbf{s} \in \mathbb{R}^n, & \text{(1c)}\end{cases}$$

où X, C et chaque A_i sont des matrices $n \times n$, $\mathcal{A}^\top \mathbf{y} = \sum_{i=1}^{k} A_i y_i$ et $X \bullet \mathbf{s}\mathbf{s}^\top = \sum_{i=1}^{n}\sum_{j=1}^{n} X_{ij} s_i s_j$. L'implication dans la dernière inégalité (1c) sert à réécrire la contrainte $X \succeq \mathbf{0}$ sous la forme d'un ensemble infini de contraintes linéaires, une pour chaque $\mathbf{s}$. On peut voir (la zone réalisable de) $\mathscr{P}$ comme un polytope si on se limite à des vecteurs s avec un nombre borné de chiffres.

Les programmes linéaires avec un nombre prohibitif de contraintes sont souvent résolus par plans coupants. Un tel algorithme construit une séquence de polytopes $\mathscr{P}_1 \supset \mathscr{P}_2 \supset \mathscr{P}_2 \ldots$ qui approximent de mieux en mieux la zone réalisable de $\mathscr{P}$. À chaque itération $\texttt{it}$ on coupe la solution optimale $\mathbf{y}_{\text{out}} = \texttt{opt-sol}(\mathscr{P}_{\texttt{it}})$ en appelant le sous problème de séparation. La séparation doit déterminer si la matrice $C - \mathcal{A}^\top \mathbf{y}_{\text{out}}$ est SDP ou pas, en calculant sa plus petite valeur propre. L'algorithme des plans coupants le plus standard est – je pense – impuissant sur ce problème SDP, d'où le vide presque absolu de travaux qui ont l'ont abordé de cette façon.

Ce papier propose d'aborder ce problème en utilisant des plans coupants projectifs, ou `Projective Cutting-Planes` [1]. L'idée de base et de remplacer la séparation avec une projection ou intersection : démarrer d'un point intérieur $\mathbf{y}_{\texttt{it}} \in \mathscr{P}$ et avancer le long d'une direction $\mathbf{d}_{\texttt{it}}$ jusqu'au point de contact avec la zone réalisable.



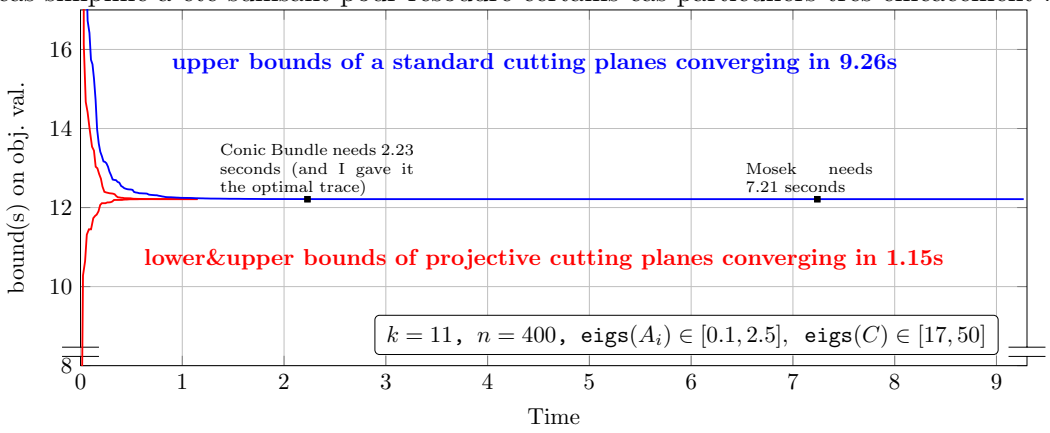
Cette figure montre un exemple d'exécution sur un PL à deux variables. À l'itération 1, `Projective Cutting-Planes` projette le point $\mathbf{y}_1 = \mathbf{0} = [0\ 0]^\top$ vers une première direction $\mathbf{d}_1$, voir la flèche noire pointillée. Cela permet de découvrir un point de contact $\mathbf{y}_1 + t_1^*\mathbf{d}_1$ ainsi que la contrainte noire. L'itération $\texttt{it} = 2$ projette le point $\mathbf{y}_2$ (au milieu de la flèche noire) vers la solution optimale $\texttt{opt-sol}(\mathscr{P}_1)$ de l'approximation extérieure $\mathscr{P}_1$ connue à l'itération

1 (le plus triangle le plus grand). Cette itération 2 génère un deuxième point de contact $\mathbf{y}_2 + t^*\mathbf{d}_2$ ainsi qu'une deuxième facette (ligne continue bleue) qui est ajoutées aux facettes de $\mathscr{P}_1$ pour construire $\mathscr{P}_2$. La troisième projection (en rouge) projette le point intérieur $\mathbf{y}_3$ vers $\texttt{opt-sol}(\mathscr{P}_2)$ et le processus est répété. Les solutions extérieures et intérieurs convergent vers l'optimum général.

Cet algorithme doit résoudre un sous problème de projection $t^* = \max\{t : \mathbf{y} + t\mathbf{d} \in \mathscr{P}\}$ à chaque itération. Dans le cadre de notre programme SDP (1a)–(1c), la solution $\mathbf{y}$ est associée à une matrice $X = C - \mathcal{A}^\top\mathbf{y}$ qui doit être SDP parce $\mathbf{y}$ est réalisable. La direction $\mathbf{d}$ peut être arbitraire (dans $\mathbb{R}^k$) et la matrice $D = C - \mathcal{A}^\top\mathbf{d}$ n'est pas forcement SDP. Le succès de `Projective Cutting-Planes` dépend de la résolution efficace de ce sous-problème de projection dans le cône SDP :

$$t^* = \max\{t : X + t \cdot D \succeq \mathbf{0}\} \tag{2}$$

Ce sous-problème est bien facile si $X \succ \mathbf{0}$. Dans ce cas là, il existe une unique factorisation Cholesky $X = KK^\top$ avec un facteur K inversible. Ainsi, on peut déterminer une unique matrice D' tel que $D = KD'K^\top$. En écrivant $X + t \cdot D = KI_nK^\top + t \cdot KD'K^\top$, cette matrice est congruente (voir, *e.g.*, Prop 1.2.3 of [2]) avec $I_n + t \cdot D'$. Cette propriété suffit pour conclure que ces deux matrices ont le même statut SDP. La projection revient ainsi à calculer $t^* = \max\{t : I_n + tD' \succeq \mathbf{0}\}$ et la solution de ce dernier programme est $t^* = -\frac{1}{\lambda_{\min}(D')}$. La projection (2) est beaucoup plus difficile si X n'est pas strictement SDP (*i.e.*, si $X \succeq \mathbf{0}$ mais $X \not\succ \mathbf{0}$), entre autres parce que la relation de congruence ci-dessus ne tient plus. Cependant, ce cas simplifié a été suffisant pour résoudre certains cas particuliers très efficacement :



Vous trouverez un papier soumis de 12 pages à `http://cedric.cnam.fr/~porumbed/papers/sdpinter.pdf` ; tous les cas de figures de projection y sont abordés, *i.e.*, le sous-problème (2) a été résolu pour tout $X \succeq \mathbf{0}$ ou $D \in \mathbb{R}^n$. L'algorithme final est très « lightweight » comparé aux méthodes de point intérieur de `mosek` ou aux idées de type Conic-Bundle. Il arrive jusqu'à résoudre des problèmes avec $n = 5000$, une taille apparemment trop grande pour d'autres algorithmes SDP. Par contre, le nouvel algorithme sera peut-être dominé par la concurence pour $k \geq 1000$ à cause d'un trop grand PL à résoudre de manière répétitive.

Références

[1] Daniel Porumbel Projective Cutting-Planes SIAM Journal on Optimization, 30(1) : 1007-1032, 2020.

[2] Daniel Porumbel Demystifying the characterizations of SDP matrices in mathematical programming, unpublished expository work around 100 pages, arXiv :2210.13072