



**HAL**  
open science

# Du sous-problème de séparation vers le sous-problème d'intersection dans les algorithmes de plan coupants

Daniel Cosmin Porumbel

## ► To cite this version:

Daniel Cosmin Porumbel. Du sous-problème de séparation vers le sous-problème d'intersection dans les algorithmes de plan coupants. ROADEF 2018, Feb 2018, Lorient, France. <hal-03956873>

**HAL Id: hal-03956873**

**<https://hal.science/hal-03956873v1>**

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.



HAL Authorization

# Du sous-problème de séparation vers le sous-problème d'intersection dans les algorithmes de plan coupants

Daniel Porumbel<sup>1</sup>

CEDRIC-CNAM, 292 rue Saint Martin, 70003 Paris, [daniel.porumbel@cnam.fr](mailto:daniel.porumbel@cnam.fr)

**Mots-clés :** *sous-problème de séparation, plans coupants*

## 1 Introduction : la séparation et les plans coupants

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

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

Pour résoudre ce type de PL, on fait souvent appel à un algorithme de plans coupants (ou **Cutting-Planes**) qui ajoute les contraintes une par une, à l'aide d'un sous-problème de séparation. Il existe plusieurs variantes de **Cutting-Planes**. Une des plus répandue 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**. La solution optimale  $\mathbf{r}$  de ce PL réduit représente l'entrée du sous-problème de séparation, qui cherche un plan coupant pour séparer  $\mathbf{r}$ .

## 2 Pour quoi opter pour un sous-problème d'intersection ?

Illustrons les idées principales par un exemple simple mais bien révélateur. Supposons que  $\mathbf{r} = [1 \ 1 \ 1]^\top$  soit solution optimale à l'itération courante où l'algorithme de plans coupants doit choisir une des contraintes suivantes :

(1)  $4x_1 + 6x_2 + 8x_3 \geq 20$ , c.-à.d.,  $\mathbf{a}^1 = [4 \ 6 \ 8]^\top$  and  $b^1 = 20$ ;

(2)  $x_1 + 2x_2 + x_3 \geq 5$ , c.-à.d.,  $\mathbf{a}^2 = [1 \ 2 \ 1]^\top$  and  $b^2 = 5$ .

Le sous-problème de séparation standard choisirait le plus souvent (1), par exemple en résolvant  $\max_{(\mathbf{a}, b) \in \mathcal{C}} b - \mathbf{a}^\top \mathbf{r}$ . Il est cependant facile d'argumenter que la contrainte (2) est plus forte que la

(1). Il suffit de normaliser les contraintes, pour obtenir des contraintes équivalentes :

(1')  $0.2x_1 + 0.3x_2 + 0.4x_3 \geq 1$ , c.-à.d.,  $\mathbf{a}^1 = [0.2 \ 0.3 \ 0.4]^\top$  and  $b = 1$ ;

(2')  $0.2x_1 + 0.4x_2 + 0.2x_3 \geq 1$ , c.-à.d.,  $\mathbf{a}^2 = [0.2 \ 0.4 \ 0.2]^\top$  and  $b = 1$ .

Si les contraintes avaient été écrites de cette manière dès le début, le sous problème de séparation aurait choisi (2'). On peut dire que (2') est une contrainte plus forte, plus violée par  $\mathbf{r}$ , car  $(\mathbf{a}^2)^\top \mathbf{r} < (\mathbf{a}^1)^\top \mathbf{r}$ . Il est facile d'argumenter que c'est mieux de normaliser les contraintes pour pouvoir les comparer de manière non biaisée.

Le sous-problème d'intersection revient à normaliser les contraintes avant d'appeler le sous problème de séparation. Il demande de résoudre

$$t^* = \max_{(\mathbf{a}, b) \in \mathcal{C}} \frac{b}{\mathbf{a}^\top \mathbf{r}} \quad (\star)$$

S'il doit choisir entre les contraintes (1) et (2), il choisit (2), car  $\frac{20}{4+6+8} = 1 + \frac{1}{9} < 1 + \frac{1}{4} = \frac{5}{1+2+1}$ .

### 2.1 Des bornes supérieures avant la fin de la convergence

Le plus souvent, un algorithme de plans coupants basé sur le sous-problème de séparation ne peut pas trouver de solution réalisable (borne supérieure) avant la fin de la convergence.

Un avantage important du sous-problème d'intersection est le fait qu'il génère assez simplement une solution réalisable  $t^*\mathbf{r}$  à chaque itération. En effet, l'équation  $(\star)$  ci-dessus garantit que  $\mathbf{a}^\top(t^*\mathbf{r}) \geq b \forall (\mathbf{a}, b) \in \mathcal{C}$ . En effet, s'il y avait une contrainte  $(\mathbf{a}_i, b_i) \in \mathcal{C}$  tel que  $\mathbf{a}_i^\top(t^*\mathbf{r}) < b_i$ , le sous problème d'intersection aurait trouvé un résultat  $\frac{b_i}{\mathbf{a}_i^\top \mathbf{r}} > t^*$  in  $(\star)$ . La figure ci-contre est révélatrice des raisons pour nommer  $(\star)$  un sous-problème d'intersection. Les bornes supérieures pourraient être très utiles par exemple dans un algorithme de **branch-and-bound**, pour élaguer des nœuds de l'arbre de recherche.

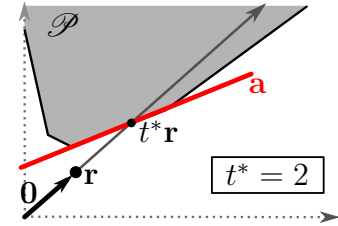


FIG. 1 – Un appel de sous-problème d'intersection qui génère la solution réalisable  $2\mathbf{r}$ .

## 2.2 La difficulté algorithmique du sous-problème d'intersection

Le sous-problème d'intersection demande de maximiser un ratio  $(\star)$ . Il est assez clair que cela est sans doute plus difficile que de maximiser un objectif linéaire comme  $\max_{(\mathbf{a}, b) \in \mathcal{C}} b - \mathbf{a}^\top \mathbf{r}$  dans le sous-problème de séparation. Cependant, il existe des cas où il est possible de résoudre le sous-problème d'intersection avec un algorithme de même complexité que l'algorithme de séparation. J'ai étudié deux contextes où le sous-problème d'intersection est aussi facile que celui de séparation : (a) la génération de colonnes [2] et (b) la décomposition de Benders [3].

Par rapport à (a), il est connu que la génération de colonnes résout un PL dual avec un nombre exponentiel de contraintes; les deux sous-problèmes (intersection ou séparation) peuvent souvent se résoudre par programmation dynamique [2]. En utilisant la programmation dynamique, le fait qu'on optimise un ratio ne complique pas la résolution, car il suffit de renvoyer l'état associé au ratio le plus élevé. Pour (b), le sous-problème de séparation demande de maximiser un PL sur un polytope de Benders. Le sous-problème d'intersection  $(\star)$  demande de résoudre un Programme Linéaire-Fractionnaire (PLF) sur le même polytope de Benders. J'ai pu utiliser la transformation Charnes-Cooper [1] pour transformer le PLF en PL, et ainsi, l'algorithme d'intersection a la même complexité que l'algorithme de séparation. En plus, le modèle de décomposition de Benders demande souvent de minimiser un PL *en Nombres Entiers* (master PLNE). La résolution itérative de ce PLNE est beaucoup plus coûteuse que la résolution du sous-problème d'intersection ou séparation (qui utilise que des PLs non entiers).

## 3 Conclusions et perspectives

La contrainte d'espace ne me permet pas de présenter des tableaux avec des résultats numériques, mais je me permets de renvoyer le lecteur vers [2, 3] pour des tests expérimentaux. Le code source est disponible en ligne à [cedric.cnam.fr/~porumbed/irm/](http://cedric.cnam.fr/~porumbed/irm/) ou resp. [cedric.cnam.fr/~porumbed/benders/](http://cedric.cnam.fr/~porumbed/benders/) avec quelques exemples de tableaux de résultats.

Ces tests numériques suggèrent qu'il pourrait être bien efficace de remplacer le sous-problème de séparation par celui d'intersection, surtout lorsqu'on peut le résoudre en un temps de calcul similaire au sous-problème de séparation. Je pense que l'approche pourrait être implémentée sur encore d'autres modèles, au moins pour produire des bornes supérieures en cours de route.

## Références

- [1] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval research logistics quarterly*, 9(3-4) :181–186, 1962.
- [2] D. Porumbel. Ray projection for optimizing polytopes with prohibitively many constraints in set-covering column generation. *Mathematical Programming*, 155(1) :147–197, 2016.
- [3] D. Porumbel. From the Separation to the Intersection Sub-problem in Benders Decomposition Models with Prohibitively-Many Constraints, disponible en-ligne à [cedric.cnam.fr/~porumbed/benders.pdf](http://cedric.cnam.fr/~porumbed/benders.pdf)