



**HAL**  
open science

# L'algorithme de la direction améliorante pour la résolution de programmes linéaires

Adrien Chan-Hon-Tong

► **To cite this version:**

Adrien Chan-Hon-Tong. L'algorithme de la direction améliorante pour la résolution de programmes linéaires. 2012. hal-00722920v1

**HAL Id: hal-00722920**

**<https://hal.science/hal-00722920v1>**

Preprint submitted on 6 Aug 2012 (v1), last revised 16 Jan 2023 (v38)

**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.

# L'algorithme de la direction améliorante pour la résolution de programmes linéaires

Adrien CHAN-HON-TONG  
CEA-LIST-DIASI-LVIC  
adrienchanhonton@gmail.com

6 août 2012

## 1 Introduction

La résolution des programmes linéaires c'est à dire des problèmes de la forme  $\max_{Av \leq b} (cv)$  avec  $A$  une matrice de  $M_{M,N}(\mathbb{Q})$ ,  $b \in \mathbb{Q}^M$  et  $c \in \mathbb{Q}^N$  est un problème très étudié dans la littérature. Ce problème est un problème polynomial [3]. Mais, les meilleures bornes polynomiales connues sont relativement élevées. Notamment, il n'existe aujourd'hui aucun algorithme déterministe connu ayant seulement besoin pour produire la solution d'un nombre d'opération dans  $\mathbb{Q}$  polynomial en  $M$  et  $N$ . En effet, tous les algorithmes polynomiaux déterministes connus utilisent un nombre d'opération dans  $\mathbb{Q}$  qui dépend de  $L$  le nombre de bits nécessaire pour écrire  $A, b, c$ .

Cette dépendance en  $L$  est la conséquence d'une recherche de la solution optimale via une descente de gradient propre aux algorithmes de type "points intérieurs". À l'opposé, les méthodes qui cherchent la solution optimale en explorant la combinatoire fini des ensembles de contraintes ne dépendent que du nombre de contraintes  $M$  et du nombre de variables  $N$ . Par exemple, la méthode du simplexe [1], très efficace en pratique, qui explore l'ensemble fini des sommets de l'espace des solutions, n'est pas sensible à  $L$ . Mais cette méthode n'est pas polynomiale [2] car le nombre de sommets augmente exponentiellement avec la dimension de l'espace.

La question de l'existence d'un algorithme possédant ces deux avantages (caractère polynomial et indépendance vis à vis de  $L$ ) est une question ouverte à ce jour, et, c'est dans ce contexte que s'inscrit cet article. Nous proposons un algorithme dont on conjecture qu'il vérifie les propriétés souhaitées dans le cas où les données de départ vérifie une hypothèse de non dégénérescence.

La suite de cet article est structuré ainsi : Dans la section 2, un état de l'art des recherches d'un algorithme fortement polynomial pour résoudre les programmes linéaires est présentée. Dans la section 3, l'algorithme proposé est présenté. Dans la section 4, une discussion sur l'efficacité de l'algorithme et sur

l'hypothèse de non dégénérescence est présentée. Enfin la section 5 contient la conclusion de cet article.

## 2 État de l'art

Depuis qu'il est démontré [3] que la résolution des programmes linéaires est un problème polynomial, deux principales voies d'améliorations sont explorés pour obtenir des algorithmes plus efficace : la recherche de règle de pivotage efficace pour le simplex [4] ou l'amélioration des méthodes de type point intérieurs.

Il existe aussi des recherches qui explorent de nouvelles voies comme [5,6].

Dans [5], un algorithme purement combinatoire est donnée pour résoudre de façon polynomiale les programmes linéaires dans lesquelles le nombre de variable par contrainte est inférieur ou égal à 2.

Dans [6], un algorithme basé sur des directions améliorantes est présenté. Il a l'avantage d'être invariant aux transformations affines du système.

Cet article se place dans la recherche de nouvelles voies et réutilise l'idée présentée dans [6] des directions améliorantes. Néanmoins au lieu d'utiliser des directions améliorantes heuristiques comme dans [6], nous proposons de chercher à construire la direction améliorante optimale compte tenu de l'objectif à chaque étape de l'algorithme.

## 3 Algorithme de la direction améliorante

### 3.1 Principe

L'idée clé de l'algorithme proposé est le point suivant. On cherche à résoudre  $\max_{Av \leq b} (cv)$  avec  $A$  une matrice de  $M_{M,N}(\mathbb{Q})$  et on dispose d'un point  $u$  tel que  $A_I u \leq b$  avec  $I$  l'ensemble des indices tel que  $A_I u = b_I$  et  $A_{[[1,M]] \setminus I} u < b_{[[1,M]] \setminus I}$  (pour tout ensemble d'indice  $\Upsilon$ , on note  $A_\Upsilon$  est la matrice dans laquelle on n'a conservé que les lignes dont l'indice est dans  $\Upsilon$ , et pour tout  $a, b \in \mathbb{N}$  on note  $[[a, b]] = \{a, a + 1, \dots, b\}$ ). Si on savait obtenir la solution  $\omega$  du problème suivant  $\max_{A_I w \leq 0, w w \leq 1} cw$  et que  $c\omega > 0$  alors  $\omega$  serait une direction d'augmentation pour notre solution courante  $u$  puisqu'elle ne fait pas croître les contraintes immédiatement saturées et améliore le résultat. Plus précisément, soit  $c\omega \leq 0$  et  $u$  (la solution courante) est la solution optimale. Soit  $\forall i \in [[1, N]] A_i \omega \leq 0$  et le problème n'est pas bornée. Soit on peut améliorer strictement  $u$  en  $u + \min_{i \in [[1, N]] | A_i \omega > 0} \left( \frac{b_i - A_i u}{A_i \omega} \right) \omega$ .

Ainsi résoudre  $\max_{Av \leq b} (cv)$  revient à résoudre un certain nombre de fois un problème de la forme  $\max_{A_I w \leq 0, w w \leq 1} cw$  (le nombre d'itération nécessaire pour converger semble être majoré par  $2M$ ).

Cela dit, obtenir directement la solution  $\omega$  du problème  $\max_{A_I w \leq 0, ww \leq 1} cw$  est un problème plus dur que le problème de départ car c'est un problème quadratique et non linéaire. Cependant, on va montrer qu'en exploitant à la fois la structure propre aux problèmes de la forme  $\max_{Qw=0, ww \leq 1} cw$  avec  $Q$  une matrice, et d'autre part en maintenant un ensemble de contrainte comme dans l'algorithme du simplexe, on peut résoudre efficacement le problème.

### 3.2 Projection sur un hyperplan

Remarquons d'abord que la solution du problème  $\max_{Qw=0, ww \leq 1} cw$  avec  $Q$  une matrice est simplement la projection de  $c$  sur le sous-espace vectoriel  $Qw = 0$  : soit  $\nu_1, \dots, \nu_R$   $R$  vecteurs obtenus en orthonormalisant les vecteurs  $Q_1, \dots, Q_N$  (par la méthode de Gram-Schmidt) alors la projection de  $c$  sur  $\{v \in E | Qv = 0\}$  est  $c - \sum_{r=1}^R (c \cdot \nu_r) \nu_r$ . En effet, pour tous vecteurs  $w$ ,  $Qw = 0 \Leftrightarrow Q'w = 0$  avec  $Q' = \begin{pmatrix} \nu_1 \\ \vdots \\ \nu_R \end{pmatrix}$  ce qui permet de se ramener à calculer la projection de  $c$  sur le sous-espace vectoriel  $Q'w = 0$ .

Néanmoins, la forme des problèmes que l'on veut résoudre est  $\max_{Qw \leq 0, ww \leq 1} cw$  et non  $\max_{Qw=0, ww \leq 1} cw$  et cela change totalement la nature du problème : il est bien connu qu'obtenir la solution du programme linéaire  $\max_{Av \leq b} (cv)$  revient à déterminer  $I^*$  tel que la solution  $p$  du problème vérifie  $A_{I^*} p = b_{I^*}$  (dans le cas où  $\forall i \in [[1, M]] A_i$  non colinéaire à  $c$ ). Ainsi le fait de savoir résoudre  $\max_{Qw=0, ww \leq 1} cw$  ne permet pas directement de résoudre  $\max_{Qw \leq 0, ww \leq 1} cw$  car trouver l'ensemble d'indice  $J^*$  tel que la solution  $\omega$  vérifie  $Q_{J^*} \omega = 0$  est aussi difficile que le problème de départ.

Mais, l'idée clé de la méthode du simplexe consiste justement à trouver l'ensemble  $I^*$  solution en maintenant un ensemble  $I$  à jours correspondant à une solution de base et à faire des échanges d'indice pour parcourir les différentes solutions de base ou géométriquement les sommets du polygone.

Notre méthode consiste à utiliser cette idée de maintenir un ensemble, mais, au lieu de maintenir un ensemble de contraintes correspondantes à un sommet, on maintient un ensemble de contraintes correspondant à une direction d'amélioration.

### 3.3 Entrée et sortie de contrainte

On cherche à résoudre  $\max_{Av \leq b} (cv)$  avec  $A$  une matrice de  $M_{M,N}(\mathbb{Q})$  et on dispose d'un point  $u$  tel que  $Au \leq b$  avec  $I$  est l'ensemble des indices tel que  $A_I u = b_I$  et  $A_{[[1, M]] \setminus I} u < b_{[[1, M]] \setminus I}$  et on connaît  $\omega$  solution de  $\max_{A_I w \leq 0, ww \leq 1} cw$

---

**Algorithme 1** Algorithme de la direction améliorante.

---

$F(A, b, c, u, I, \omega, J)$  :

$u \leftarrow \left( u + \min_{i \in [[1, N]] | A_i \omega > 0} \left( \frac{b_i - A_i u}{A_i \omega} \right) \omega \right)$   
 $r$  est la contrainte tel que  $\left( \frac{b_r - A_r u}{A_r \omega} \right)$  est minimal  
 mettre à jour de  $I$  :  $A_I u = b_I$  et  $A_{[[1, M]] \setminus I} u < b_{[[1, M]] \setminus I}$   
 calculer les solutions des  $|J| + 1$  problèmes  $\max_{A_{J \setminus \Theta \cup \{r\}} w = 0, w w \leq 1} cw$  avec  $|\Theta| \leq 1$   
 chercher parmi ces solutions, la solution de  $\max_{A_I w \leq 0, w w \leq 1} cw$   
 si aucune solution n'est trouvée ou si  $c\omega \leq 0$   
 renvoyer  $u$   
 si  $\forall i \in [[1, N]] A_i \omega \leq 0$   
 renvoyer "le problème n'est pas borné"  
 mettre à jour  $J$  :  $A_J \omega = 0$  et  $A_{I \setminus J} u < 0$   
 renvoyer  $F(A, b, c, u, I, \omega, J)$

Précondition :  $Au \leq b$  avec  $I$  l'ensemble tel que  $A_I u = b_I$  et  $A_{[[1, M]] \setminus I} u < b_{[[1, M]] \setminus I}$  et  $\omega$  solution de  $\max_{A_I w = 0, w w \leq 1} cw$  avec  $J$  l'ensemble tel que  $A_J \omega = 0$  et

$A_{[[1, M]] \setminus J} \omega < 0$

Post-condition : retourne  $u$  solution de  $\max_{Av \leq b} (cv)$

---

et  $J$  l'ensemble tel que  $A_J \omega = 0$   $A_{I \setminus J} \omega < 0$  (on supposera que  $c\omega > 0$  et  $\exists i \in [[1, N]], A_i \omega > 0$ ). Alors  $u$  peut être amélioré en  $u + \min_{i \in [[1, N]] | A_i \omega > 0} \left( \frac{b_i - A_i u}{A_i \omega} \right) \omega$ ,

ce faisant  $I$  devient  $I \setminus \Gamma_- \cup \Gamma_+$ ,  $\Gamma_-$  étant les contraintes sortantes et  $\Gamma_+$  les contraintes entrantes. Étant donnée qu'on se déplace sur l'ensemble du polyèdre (et pas les sommets), on peut supposer (hypothèse de non dégénérescence) que  $|\Gamma_+|$  (le cardinal de  $\Gamma_+$ ) est 1. C'est à dire qu'une seule nouvelle contrainte d'indice  $r$  se rajoute. Cette contrainte se rajoute aussi à  $J$  puisque si on bute sur la contrainte  $r$ , elle sera nécessairement saturée quand on cherchera une direction améliorante  $\max_{A_I w \leq 0, w w \leq 1} cw$  (avec la valeur actualisée de  $I$ ). De même, excepté

si la nouvelle contrainte  $r$  a une orientation particulière vis à vis des contraintes précédentes, pas plus de 1 contrainte ne devrait sortir de  $J$  (par contre il est possible qu'aucune des contraintes ne sortent). Ainsi  $\omega$  doit être (hypothèse de

non dégénérescence) parmi les solutions des problèmes  $\left( \max_{A_{J \setminus \Theta \cup \{r\}} w = 0, w w \leq 1} cw \right)$

avec  $|\Theta| \leq 1$ .

Ainsi dans les cas de non dégénérescence, on peut faire évoluer  $J$  de façon simple et ainsi résoudre le programme linéaire uniquement en effectuant des projections et sans avoir à résoudre de problème quadratique. Cela est résumé dans l'algorithme 1.

## 4 Discussion

### 4.1 Analyse de l'algorithme

Dans cet article, on ne donne ni de démonstration du fait que l'algorithme de la direction améliorante retourne la solution du problème ni du fait que l'algorithme termine en temps polynomial. À vrai dire ces deux propriétés ne sont pas vérifiées dans le cas général. Néanmoins, il semble que ces deux propriétés sont vraies dans les cas non dégénérés c'est à dire dans les cas où à chaque étape de l'algorithme une seule contrainte d'indice  $r$  se rajoute dans  $J$  et où la solution optimale du sous-problème  $\max_{A_I w \leq 0, w w \leq 1} cw$  se trouve parmi les solutions qui saturent  $J \setminus \Theta \cup \{r\}$  avec  $\Theta$  un ensemble contenant au plus 1 élément ( $I$  représentant les contraintes saturées au point courant et  $J$  les contraintes saturées par la direction d'amélioration précédente : voir l'algorithme 1).

En effet dans les cas non dégénérés la valeur retournée est bien la solution du problème initial car la variable  $\omega$  est la solution de  $\max_{A_I w \leq 0, w w \leq 1} cw$  où  $I$  est l'ensemble des contraintes saturées en  $u$  donc si  $c\omega \leq 0$  c'est qu'il n'existe aucune direction d'amélioration en  $u$ . Or les programmes linéaires vérifient qu'en tout point non optimal, il existe une direction d'amélioration.

De plus, dans les cas non dégénérés, le nombre itérations semble être borné par  $2M$  car si  $\omega$  est bien la meilleur des directions d'amélioration à chaque étape de l'algorithme alors une contrainte ne peut apparaître qu'une fois dans  $J$  et disparaître de  $J$  qu'une seule fois. Or chaque appel récursif de la fonction effectue au plus  $O(M^3N)$  opération sur  $\mathbb{Q}$  ( $M$  et  $N$  étant les dimensions de la matrice  $A$ ).

Ainsi dans les situations non dégénérés, l'algorithme résolve un programme linéaire de façon fortement polynomial en  $O(M^5N)$  opérations dans  $\mathbb{Q}$ . Ainsi couplé avec des algorithmes efficaces pour réaliser les opérations dans  $\mathbb{Q}$  (FFT pour la multiplication, Newton pour la division), la complexité temporelle de l'algorithme est en  $O(M^5NL)$  contre  $O(L^2)$  pour les algorithmes polynomiaux déterministes connus.

### 4.2 Hypothèse de non dégénérescence

L'hypothèse de non dégénérescence est restrictive. Néanmoins, des tests préliminaires soutiennent qu'en perturbant de façon aléatoire chaque coefficients, on obtient des données sur lesquelles cette hypothèse semble être vérifiée. Plus précisément, il semble que dans les cas où la solution est faiblement modifiée par une faible perturbation sur les données du problème, alors l'hypothèse de non dégénérescence s'applique.

Ainsi, même si cet article ne propose pas un nouvel algorithme complet pour résoudre les problèmes linéaires, il apporte un nouvel élément de réflexion dans la recherche d'un algorithme fortement polynomial pour résoudre les programmes linéaires.

### 4.3 Initialisation de l'algorithme

L'algorithme fonctionne en mettant à jour un ensemble  $J$  qui correspond aux contraintes saturées par la direction d'amélioration précédente. Le problème du calcul du premier ensemble  $J$  n'est pas résolu dans cet article. Néanmoins, de la même façon que quelque soit le programme linéaire  $\max_{Av \leq b} (cv)$  il existe un programme linéaire  $\max_{A'v \leq b'} (c'v)$  tel que  $0 \in \{v \in E | A'v \leq b'\}$  et qui soit équivalent (ie de taille équivalente et dont la résolution donne une solution au premier problème), il est possible qu'il soit possible de transformer un programme linéaire quelconque en un autre programme linéaire équivalent et possédant à la fois un point trivial et une première direction d'amélioration triviale.

## 5 Conclusion

Dans cet article, un algorithme est présenté dans le contexte de la programmation linéaire. Dans le cas où les données d'entrée de l'algorithme vérifient une hypothèse de non dégénérescence, on conjecture que cet algorithme retourne la solution du programme linéaire en effectuant un nombre d'opérations dans  $\mathbb{Q}$  polynomial en le nombre de contraintes et le nombre de variables. Cet algorithme représente potentiellement une avancée majeure par rapport aux algorithmes polynomiaux actuelles ayant besoin d'un nombre d'opérations dans  $\mathbb{Q}$  polynomial en le nombre de contraintes et le nombre de variables et linéaire en la taille en bits de l'entrée.

## Références

- [1] G. B. Dantzig., "Maximization of linear function of variables subject to linear inequalities.", 1951
- [2] V. Klee et G. J. Minty, "How good is the simplex algorithm ?", 1972
- [3] N. KARMARKAR, "A NEW POLYNOMIAL-TIME ALGORITHM FOR LINEAR PROGRAMMING", 1984
- [4] Kelner, J.A. et Spielman, D.A., "A randomized polynomial-time simplex algorithm for linear programming", 2006
- [5] N Megiddo, "TOWARD A GENUINELY POLYNOMIAL ALGORITHM FOR LINEAR PROGRAMMING", 1981
- [6] Mihály Bãrãsz Santosh Vempala, "A new approach to strongly polynomial linear programming", 2010