



HAL
open science

Could interior point algorithm for linear programming can be helped by greedy improvement ?

Adrien Chan-Hon-Tong

► **To cite this version:**

Adrien Chan-Hon-Tong. Could interior point algorithm for linear programming can be helped by greedy improvement ?. 2019. hal-00722920v24

HAL Id: hal-00722920

<https://hal.science/hal-00722920v24>

Preprint submitted on 11 Jan 2020 (v24), 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.

Could interior point algorithm for linear programming can be helped by greedy improvement ?

Adrien CHAN-HON-TONG

January 11, 2020

Abstract

The structure of main polynomial time algorithms for linear program forbid them to use greedy improvement which could allow to stick to the combinatorial structure of the problem.

This paper reviews interior point for linear feasibility, and, wonder if a direct optimization could be feasible.

1 Introduction

Linear programming is the very studied task of optimizing a linear criterion under linear equality and inequality constraints. This problem has been first tackled by exponential algorithms like simplex [2] or perceptron [6]. Today, this problem is tackled by polynomial time algorithms like ellipsoid method [4, 3], log barrier method [5], or recently, Chubanov method [1].

1.1 Notation

The set of matrices of size $M \times N$ on \mathbb{R} is written $\mathbb{R}^{M \times N}$ (the same for \mathbb{Q} or \mathbb{Z}). This set is a vectorial space with an addition written $+$, and, a product between a scalar and a matrix written \times . If A is a matrix from $\mathbb{R}^{M \times N}$, then, the transposed matrix is written $A^T \in \mathbb{R}^{N \times M}$. The set of all matrices of any size has an inner product which is represented by juxtaposition of matrices: if $A \in \mathbb{R}^{I \times J}$, and, $B \in \mathbb{R}^{J \times K}$, then, AB is the matrix product of A and B which is in $\mathbb{R}^{I \times K}$. If A is a matrix, then A_i is the row i of A considered as a $1 \times N$ matrix. If I is the set of indexes, A_I is the submatrix when keeping only row i from i . Vectors of dimension N are considered as matrix with size $N \times 1$, but, matrices 1×1 are considered as real: so, if v is a vector v_i is the value of component i . $\mathbf{0}$, $\mathbf{1}$ are the vectors with all components being 0 or 1, and, \mathbf{b}_i the i vector of the natural basis i.e. all components are 0 except component i with is 1. Finally, $Det(A)$ is the maximal determinant of any square submatrix of A (with maximal number of columns).

Greek letter are used to design abstract object (object which could not exist when proving contradiction, optimal solution which are not computable...).

1.2 Equivalence on linear programming

The native form of linear program is the task of solving $\min_{x \in \mathbb{Q}^N / Ax \geq b} cx$ for given $A \in \mathbb{Z}^{M \times N}$ a matrix, $b \in \mathbb{Z}^M$ and $c \in \mathbb{Z}^N$ some vectors.

But it is well known that solving $\min_{x \in \mathbb{Q}^N / Ax \geq b} cx$ is equivalent to find a solution $y \in \mathbb{Q}^{M+N}$ (or prove there is not) to a system of inequality $Hy \geq h$ with $(H \in \mathbb{Z}^{(M+N+2) \times (N+M)}$ and $h \in \mathbb{Z}^{N+M+2})$ due to primal dual theory. There is a solution if and only if the original problem is both feasible and bounded. As finding (or prove there is not) y such that $Hy \geq h$ is equivalent to solve $\min_{z, t \in \mathbb{Q}^N \times \mathbb{Q} / Hz + t \geq h, t \geq 0} t$ which is a feasible and bounded problem, it

is possible to consider primal dual again (on this last one). Thus, solver can assume that some solution exists, and, that the task is to find it. Then, as maximal determinant of a sub matrix is a polynomial in the binary size of the matrix, any solution of $Hz + t \geq h, t \geq 0, -t \geq -\varepsilon$ for a decidable ε can be converted into a solution by greedy projection. So, finding y (which exists) such that $Hy \geq h$ is equivalent to solve a system of strict inequality $Gz > g$. Finally, $Gz > g$ is equivalent to find x (not related to original problem) such that $Ax > \mathbf{0}$ because $\begin{pmatrix} G & -g \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ t \end{pmatrix} > \mathbf{0}$. This is also trivially equivalent to produce x such that $Ax \geq \mathbf{1}$.

This way, any linear programming solver can assume without restricting the generality to expect an input $A \in \mathbb{Z}^{M \times N}$, with the task of producing $x \in \mathbb{Q}^N$ such that $Ax > \mathbf{0}$, with the prior that such x exists.

Then, it is classical that one could remove extra variables until kernel of A ($Ker(A) = \{x \in \mathbb{Q}^N / Ax = \mathbf{0}\}$) is reduce to $\mathbf{0}$: if there exists x, χ such that $A\chi = \mathbf{0}$, $\chi \neq \mathbf{0}$ and $Ax > \mathbf{0}$, then, $A(x - \frac{x\chi}{\chi\chi}\chi) = Ax > \mathbf{0}$, and, $\chi(x - \frac{x\chi}{\chi\chi}\chi) = 0$. So, one could just find y such that $Ay > \mathbf{0}$ after injecting $y\chi = \mathbf{0}$ (i.e. $x_1 = -\frac{1}{\chi_1} \sum_{n=\{2, \dots, N\}} \chi_n x_n$ into $Ay > \mathbf{0}$ - this does not change the form of the problem).

Finally, if I, J is a partition of all indexes of rows of A , then, finding $A_J x > \mathbf{0}$, $A_I x = \mathbf{0}$, and, $A_I y > \mathbf{0}$ is sufficient to find z such that $Az > \mathbf{0}$. Indeed, even if $A_J y$ may not be acceptable positive, it will be possible to deal with it by adding λx which does not modify $A_I y$. So if one find x , then it is only required to look for y (this strictly reduce the number of rows, and, possibly the number of variable as extra variable are removed). So, with all this reduction, one could consider reduced homogeneous linear feasibility only.

Definition: reduced homogeneous linear feasibility

Linear programming is equivalent to the task of find $x \in \mathbb{Q}^N$ such that $Ax \geq \mathbf{0}$, $Ax \neq \mathbf{0}$ given $A \in \mathbb{Q}^{M \times N}$ with $Ker(A) = \{\mathbf{0}\}$, and, with the prior that there exists y such that $Ay > \mathbf{0}$.

2 Interior point

This section describes the classical interior point method in context of homogeneous linear feasibility (precisely, usually barrier function is used on the constraint $\log(A_i x - b_i)$ to forbid $A_i x \leq b_i$ - but it requires to deal with problem in primal form, and, to update cost function c during runtime - here barrier function is only used on decomposition of x):

Let consider $F(v)$ from $]0, \infty[^M$ to \mathbb{R} , and, $x(v)$ from $]0, \infty[^M$ to \mathbb{R}^N defined by:

$$\begin{aligned} x(v) &= A^T v = \sum_{m=1}^M v_m A_m^T \\ F(v) &= x(v)^T x(v) - \mathbf{1}^T \log(v) \\ &= v^T A A^T v - \mathbf{1}^T \log(v) = \sum_{i,j=1}^M v_i v_j \times A_i A_j^T - \sum_{i=1}^M \log(v_i) \end{aligned}$$

Let write $\frac{1}{v}$ for the vector in \mathbb{R}^M such that value of component m is $\frac{1}{v_m}$, then, $\nabla_v F = Ax(v) - \frac{1}{v}$. The link with linear feasibility is that: if there exists $\nu \in]0, \infty[^M$ such that $\nabla_\nu F = \mathbf{0}$, then, $Ax(\nu) = \frac{1}{\nu} > \mathbf{0}$.

Even more, the two assumptions $\|\nabla_v F\| \leq \delta$ and $\exists k$ such that $A_k x(v) < 0$ implies that $|A_k x(v)| + \frac{1}{v_k} \leq \delta$ i.e. $v_k \geq \frac{1}{\delta}$.

But, at this point, even for very small δ , this could be possible. Currently it is possible if exists v such that $x(v)^T x(v) = v^T A A^T v = 0$. But, as y exists, then $A A^T$ is semi definite positive (because there is a solution). So, there exists $\lambda_- > 0$ the minimal not null spectral value of $A A^T$ with binary size polynomial in binary size of A such that $x(v)^T x(v) = v^T A A^T v \geq \lambda_- v^T v$ when $v \geq \mathbf{0}$.

Main consequence is that $F(v) \geq \lambda_- v^T v - \mathbf{1}^T \log(v)$.

So, let introduce the function $f(t)$ from $]0, \infty[$ to \mathbb{R} defined by $f(t) = \lambda_- t^2 - \log(t)$. This function goes to ∞ when t goes to 0 or to ∞ (t^2 dominates $\log(t)$). So, this function admits a minimum $f^* = f(\tau) = 1 + \frac{1}{2} \log(\lambda)$ which corresponds to $f'(\tau) = 0$ for $\tau = \frac{1}{\sqrt{\lambda}}$.

By combining the two previous statements, $F(v) \geq M f^*$, so F admit a minimum $F^* = M + \frac{M}{2} \log(\lambda)$ corresponding to $\|\nabla_v F\| = 0$, and importantly, $F^* = F(v)$ is bounded by a polynomial in the binary size of A .

Also, as $f(t)$ goes to ∞ when t goes to 0 or to ∞ , then, $f(t) \leq \phi$ implies $t \in [\alpha(\phi), \beta(\phi)]$. Precisely, $\alpha(\phi) = \exp(-\phi)$, because, $-\log(t) \leq f(t) \leq \phi$. For $\beta(\phi)$, either it is τ , or, β is the inverse of $f(t)$ on $[\tau, \infty[$. Currently, one can lower bound $f(t)$ by $f(\frac{2}{\sqrt{\lambda}}) + (t - \frac{2}{\sqrt{\lambda}}) f'(\frac{2}{\sqrt{\lambda}})$ (this is a very coarse lower bound but $\log(\beta)$ is polynomial in the binary size of A , so...). Now, let consider F , and, an index k . Then, $F(v) \geq \lambda_- v^T v - \mathbf{1}^T \log(v) \geq (M-1) f^* + \lambda_- v_k^2 - \log(v_k)$. So, $\lambda_- v_k^2 - \log(v_k) \leq F(v) - (M-1) f^*$. So, a fortiori $\alpha(F(v) - (M-1) f^*) \leq v_k \leq \beta(F(v) - (M-1) f^*)$ at any point and for all k .

So, it is not possible to have simultaneously the three statements $F(v) < F(v_0)$, $\|\nabla_v F\| \leq \frac{1}{\beta(F(v_0) - (M-1) f^*)}$ and $\exists k$ such that $A_k x(v) < 0$, because first

assumption implies that $v_k < \beta(F(v_0) - (M - 1)f^*)$ but last two imply that $v_k \geq \beta(F(v_0) - (M - 1)f^*)$. So, if one find v such that $F(v) < F(v_0)$, and, $\|\nabla_v F\| \leq \frac{1}{\beta(F(v_0) - (M - 1)f^*)}$, then, $Ax(v) > \mathbf{0}$.

Now, F is self concordant (a sum of -log and a quadratic term). So, damped newton method can be applied on F .

This descent has two phases. First phase is characterized by $(\nabla_v F)^T \nabla_v^2 (\nabla_v F) \geq \frac{1}{4}$ but $F(v_{after_update}) \leq F(v) - \frac{1}{4}$. So, this phase can not last more than $4(F(v_0) - F^*)$ steps.

Let consider $v_0 = (\frac{1}{\sqrt{A_1 A_1^T}}, \dots, \frac{1}{\sqrt{A_M A_M^T}})$ then $F(v_0) \leq M^2 + 2 \sum_{m=1}^M \log(A_m A_m)$

i.e. $F(v_0)$ is bounded by a polynomial in the binary size of A . So, if $(\nabla_v F)^T \nabla_v^2 (\nabla_v F) \geq \frac{1}{4}$ until abrupt convergence, then, the number of steps is lower than $4(F(v_0) - F^*)$ which is bounded by a polynomial in the binary size of A . In this case the problem is solved in polynomial time.

If the first phase ends before the convergence of the algorithm. Then, it means that after a polynomial number of steps, $(\nabla_v F)^T \nabla_v^2 (\nabla_v F) \leq \frac{1}{4}$. But, then, for every steps after this point, $(\nabla_{v_{after}} F)^T \nabla_{v_{after}}^2 (\nabla_{v_{after}} F) \leq 2((\nabla_v F)^T \nabla_v^2 (\nabla_v F))^2$. So, $\nabla_{v_{step-t}} F^T \nabla_{v_{step-t}}^2 (\nabla_{v_{step-t}} F) \leq \varepsilon$ with $t \leq O(\log \log(\frac{1}{\varepsilon}))$. But, $(\nabla_v F)^T \nabla_v^2 (\nabla_v F) \geq \exp(-2F(v_0)) \nabla_v F^T \nabla_v F$ because the hessian is AA^T (semi definite positive) plus $\frac{1}{v^2}$ a definite positive matrix bounded by $\exp(-2F(v_0))$. So, newton descent will make $\nabla_v F^T \nabla_v F \leq \frac{1}{\exp(-2F(v_0))} ((\nabla_v F)^T \nabla_v^2 (\nabla_v F))^2 \leq \varepsilon$ after $\log \log(\frac{1}{\varepsilon \exp(-2F(v_0))})$. For $\varepsilon = \frac{\lambda}{\beta(F(v_0) - (M - 1)f^*)}$, one has the solution of the problem in a number of steps polynomial in the binary size of A .

Finally, let stress, this is not done yet: applying naive damped newton method will make binary size of v becoming exponential (and require to compute the root of $1 + (\nabla_v F)^T \nabla_v^2 (\nabla_v F)$ for the damped step). So, naively, each step will require exponential binary operation in binary size of A . This is why, it is important to do the damped newton descent while keeping an acceptable binary size.

Let consider v, v', v'' 3 successive points during the newton descent. From v the descent needs K steps, $K - 1$ from v' and $K - 2$ from v'' . If one can round v'' such that $F(\text{round}(v''))$ is lower than $F(v')$, then, the descent should required less than $K - 1$ steps from $\text{round}(v'')$ i.e. if rounding ensure that $F(\text{round}(v''))$ is lower than $F(v')$, then, the algorithm will convergence is less than $2K$ steps (where K is the number of steps for the infinite precision convergence).

So, the idea is to force denominator of even-step v to be $2^{-\Gamma}$ with $2^{-\Gamma}$ sufficiently small such that the error (less than $\frac{M}{2^{-\Gamma}}$) between native and rounded vectors does not make F to go higher than the corresponding odd-step v (numerator can be arbitrary large (as $v_k \leq \beta(F(v_0) - (M - 1)f^*)$), numerator can not be higher than $2^{\Gamma + \text{ceil}(\log(\beta(F(v_0) - (M - 1)f^*)))}$).

Hopefully, v is upper and lower bounded, then, F is lipschitz. Here, F is even lipschitz with a coefficient γ polynomial in the binary size of A . So, it is sufficient that $2^{M\gamma - \Gamma}$ is lower than the smallest newton improvement. Let ω be

the smallest improvement. It is sufficient that $\Gamma \geq M\gamma - \log(\omega)$. And, ω is $\frac{1}{4}$ for first phase, and, related to the gradient in the second phase (so higher than ε from which one has created a solution).

3 Helping interior point

Frustratingly, neither ellipsoid method, or, the interior point described above can take advantage of easy greedy improvement.

In ellipsoid method, even if $A_i x \ll 0$ adding A_i to x except of sticking to the global scheme is a bad idea because it will require to increase the bounding ellipsoid to be sure not to forgot solution.

In the interior point method, it is possible to check if moving along A_i decreases F . Yet, as F implies log function, it is not possible to have infinite precision in this computation.

Also, the method is quite complex compared to exponential simplex (especially, it leads to numerical issue which can be solved theoretically by rounding one vector each two, but, may a coding nightmare).

Yet, instead of relying on F . One could wonder if it is not possible to directly minimize $x(v)^T x(v)$ under constraint $v \geq \mathbf{0}$, $v^T v \geq 1$. Let stress that this is very different from minimizing $x(v)^T x(v)$ under constraint $Ax(v) \geq \mathbf{1}$: constraints are linked in $Ax(v) \geq \mathbf{1}$, but, independent in $v \geq \mathbf{0}$ - only $v^T v = 1$ links the variable. Currently, the resulting solution may have nothing to do with a support vector machine solution related to A .

Yet, **let ν the solution of $\min_{v \geq \mathbf{0}, v^T v \geq 1} x(v)^T x(v)$, then, $Ax(\nu) \geq \mathbf{0}, Ax(\nu) \neq \mathbf{0}$** i.e. $x(\nu)$ is a solution of the homogeneous linear feasibility instance.

indeed, if $A_i x(\nu) < 0$, then, one could add so A_i^T to $x(\nu)$ (corresponding to add some \mathbf{b}_i to ν to improve the solution). Also, $x(\nu) \neq \mathbf{0}$, because, $\nu \neq \mathbf{0}$, so, $y^T x(\nu) \geq \varepsilon \mathbf{1}^T \nu$ (y is the solution i.e. such that $Ay \geq \varepsilon \mathbf{1} > \mathbf{0}$). So, $Ax(\nu) \neq \mathbf{0}$, otherwise, $x(\nu)^T x(\nu) = 0$ (it is a positive sum of row of $Ax(\nu)$).

So, one could consider the following algorithm:

Algorithm

From v such that $v_I = \mathbf{0}$, $A_I x(v) \geq \mathbf{0}$, $v_J > \mathbf{0}$, $v^T v = 1$, one could look for u solving:

$$\min_{u: u^T v = 0, u_I = \mathbf{0}} (v + u)^T A A^T (v + u) = \min_{u: u^T v = 0, u_I = \mathbf{0}} x(v + u)^T x(v + u)$$

Solving this problem can be done (approximatively) with pure linear algebra without combinatorial issue.

Then, if $v + u \geq \mathbf{0}$, $v = v + u$ or $v = v + l \times u$ with l to add some vector in I . Then, if $A_I x(v)$ is not higher than 0, some A_i^T can be added to x (i.e. \mathbf{b}_i can be added to v). Finally, v can be scaled to restore $v^T v = 1$.

End algorithm

Both steps of adding u and adding A_i^T makes $v^T v$ larger while making $x(v)^T x(v)$ decreases. So, rescaling also makes $x(v)^T x(v)$ because v is always larger than 1 just before the rescaling. So, this algorithm does not loop.

At this point, it may not converge, or, reaches a point with $u = \mathbf{0}$.

Yet, $u = \mathbf{0}$ means in reality that v is a solution, because, $A_I x(v) \geq \mathbf{0}$ and $A_J x(v) \geq \mathbf{0}$ while $v \neq \mathbf{0}$, otherwise let split J in K, H with positive and negative then u such that $u_j = \frac{\varepsilon}{|K|v_j}$ if $j \in K$, and, $u_j = \frac{\varepsilon}{|H|v_j}$ if $j \in H$ could have been considered.

Currently, when, $I = \emptyset$, this is even simpler just considering the function $f(t) = x \left(\frac{t \times \nu + (1-t) \times v}{\|t \times \nu + (1-t) \times v\|} \right)^T x \left(\frac{t \times \nu + (1-t) \times v}{\|t \times \nu + (1-t) \times v\|} \right) - f$ is strictly decreasing from 0 to 1, and so, the derivatives of $\frac{t \times \nu + (1-t) \times v}{\|t \times \nu + (1-t) \times v\|}$ is a candidate for u .

For the convergence, it is not clear that $Ax(v) \geq \mathbf{0}$, even if $x(v)^T x(v)$ is arbitrary close to the optimum. Yet, even in this case, if $Ax(v) \geq -\frac{\mu}{2Det(A)} \mathbf{1}$ one can produce a solution.

$\mu = \min_{v \geq \mathbf{0}, v^T v \geq 1} \max_i (A_i x(v))^2$ is not nul, So, if $Ax(v) \geq -\frac{\mu}{2Det(A)} \mathbf{1}$, then, $Ax(\frac{1}{\mu} \times v) \geq -2Det(A) \times \mathbf{1}$, but, there is i such that $A_i x(\frac{1}{\mu} \times v) \geq 1$. So, one can consider $\min_{x, t \quad Ax + t \times (1 - \mathbf{b}_i) \geq \mathbf{b}_i, t \geq 0} t$, this auxiliary problem can be initialized by $x(v)$ and $\frac{1}{2Det(A)}$. Just using greedy projection, one may reach a vertex with $t \leq \frac{1}{2Det(A)}$. As cramer rule apply on vertex, it means that $t = 0$ (so x is a solution of $Ax \geq \mathbf{b}_i$, and, so a solution of the original problem (notice that it is not necessarily matched with some v).

So, the offered algorithm should converges because at least u could be a simple perceptron like update $v_step_t + 1 = v_step_t - \frac{A_{i_t} x(v_step_t)}{A_{i_t} A_{i_t}^T} \mathbf{b}_{i_t}$ corresponding to $x_{t+1} = x_t - \frac{A_{i_t} x(v_step_t)}{A_{i_t} A_{i_t}^T} A_{i_t}^T$

As, this update is at least $\frac{1}{2Det(A)}$ (or one can exit to build a solution), then, the algorithm converges.

At this point the complexity is exponential, but, the offered algorithm has some interesting feature:

- it is close to a well known polynomial method (it tries to minimize $x(v)^T x(v)$ with constraints $v \geq \mathbf{0}$, and, $v^T v = 1$ instead of directly minimizing $x(v)^T x(v) - \log(v)$, so, the complexity *may be* polynomial (for example if one could match newton step with algorithm step).
- yet, this algorithm manipulated simpler object like only quadratic function: this allows for example simple check to see if using some greedy improvement does not break the algorithm convergence
- there is some combinatorial part I which appear will the algorithm is currently an interior point algorithm.

References

- [1] Sergei Chubanov. A polynomial projection algorithm for linear feasibility problems. *Mathematical Programming*, 153(2):687–713, 2015.
- [2] George B et. al. Dantzig. The generalized simplex method for minimizing a linear form under linear inequality restraints. In *Pacific Journal of Mathematics American Journal of Operations Research*, 1955.
- [3] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
- [4] Leonid Khachiyan. A polynomial algorithm for linear programming. *Doklady Akademii Nauk SSSR*, 1979.
- [5] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. Siam, 1994.
- [6] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.