An $\varepsilon$-Precise Feasible Solution to a Linear Program with a Convexity Constraint in $1/\varepsilon^2$ Iterations Independent of Problem Size

by

George B. Dantzig

TECHNICAL REPORT SOL 92-5

October 1992

Research and reproduction of this report were partially supported by the National Science Foundation Grants DMS-8913089, ECS-8906260; the Department of Energy Grant DE-FG03-92ER5116 and the Electric Power Research Institute Grant RP8010-09.

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do NOT necessarily reflect the views of the above sponsors.

Also issued as Operations Research Department Technical Report 92-14. Reproduction in whole or in part is permitted for any purposes of the United States Government. This document has been approved for public release and sale; its distribution is unlimited.
An $\epsilon$-Precise Feasible Solution to a Linear Program with a Convexity Constraint in $1/\epsilon^2$ Iterations Independent of Problem Size

by George B. Dantzig

Abstract

Von Neumann, in a private communication to the author in 1948, proposed the first interior algorithm for finding a feasible solution to a linear program with a convexity constraint. I prove it has the remarkable property that, independent of the number of rows $m$ and columns $n$, it generates in less than $1/\epsilon^2$ iterations a feasible solution with a precision $\epsilon$ (where $\epsilon^2$ is the sum of the squares of errors of fit of the left hand side of the equations to the right hand side) when the general problem is recast in the form:

$$x \geq 0, \quad \sum_{j=1}^{n} x_j = 1, \quad \sum_{j=1}^{n} P_j x_j = 0, \quad \|P_j\|_2 = 1 \quad \forall j.$$  

Letting $0 < \delta \leq 1$, denote the non-zero coefficient density, the work per iteration is $\delta mn + 2m + n + 9$ multiplications and $\delta mn + m + n + 9$ additions or comparisons.

Defining the problem as solved if a solution is obtained with precision $\epsilon$, say $\epsilon = 10^{-10}$, the algorithm's bound on the number of arithmetic operations to attain this precision is a polynomial expression of degree $2$ which is $1.5$ degrees lower than that of Karmarkar's interior algorithm, [1]. Nevertheless, for the bound on the number of arithmetic operations iterations to be lower than that of the Karmarkar bound, the number of variables has to be very large, for example for a precision $\epsilon = 10^{-10}$, it has to be greater than $1.3 \times 10^{18}$. I reproduce von Neumann's algorithm and my proof of convergence, which dates back to my original correspondence with von Neumann; to the best of my knowledge, neither has ever been published.
Introduction

The general linear program feasibility problem with a convexity constraint

(1) find \( y = (y_1, y_2, \ldots, y_n) \geq 0, \quad \sum_1^n Q_j y_j = b, \quad \sum_1^n y_j = 1, \quad Q_j \in \mathbb{R}^n \)

can be reduced to the "standard" form:

(2) find \( x = (x_1, x_2, \ldots, x_n) \geq 0, \quad \sum_1^n P_j x_j = 0, \quad \sum_1^n x_j = 1, \quad P_j \in \mathbb{R}^n \)

where \( \|P_j\|_2 = 1 \; \forall j \). This is done by letting

(3) \( \tilde{Q}_j = Q_j - b, \quad \tilde{P}_j = \tilde{Q}_j / \|\tilde{Q}_j\| \).

and noting that if \( x = x^o \) solves (2) then \( y = y^o \) solves (1) where

\[ y^o_j = (x^o_j / \|\tilde{Q}_j\|) \sum_1^n (x^o_k / \|\tilde{Q}_k\|). \]
The von Neumann Algorithm

Geometrically, see figure, the columns $P_j$ in $\mathbb{R}^m$ can be viewed as points lying on a hypersphere $S_1$ with radius $= 1$ and center at the origin $0$. The problem then becomes one of assigning nonnegative weights $x_j$ to the points $P_j$ so that their weighted center of gravity is the origin $0$.

**Initiation $t = 1$:**

The algorithm can be initiated with any approximation to the origin: $A^1 = \sum_j P_j x_j^1$, $\sum_j x_j^1 = 1$, $x_j^1 \geq 0$ where $x = x^1$ is arbitrary. To be specific, set

\begin{equation}
(4) \quad x_1^1 := 1, \quad x_j^1 := 0 \text{ for } j \neq 1, \quad A^1 = P_1, u_1 := \|A_1\| = 1; \quad t := 2
\end{equation}

**Iteration $t \geq 2$:**

At the start of iteration $t \geq 2$, one is given an approximate solution:

$z = z^{t-1} \geq 0$, $\sum_j z_j^{t-1} = 1$, $A^{t-1} = \sum_j P_j z_j^{t-1}$, $u_{t-1} = \|A^{t-1}\|$.

**Step 1:** Among directions from the origin to $P_j$, find the direction $P_s = P_j$ which makes the largest angle with the direction $A^{t-1}$:

$s := \text{ARGMIN}_j (A^{t-1})^T P_j$;

**Step 2:** Let $v = (A^{t-1})^T P_s$. If $v > 0$, then all points $P_j$ lie on one side of the hyperplane through the origin perpendicular to the direction $A^{t-1}$ implying the linear program is infeasible since clearly no convex combination of the points $P_j$ can be found having the origin as center of gravity.

STOP IF $v > 0$;
Step 3: Choose as next approximate $A^t$ the closest point to the origin on the line segment joining $A^{t-1}$ to $P_s$. Update:

\begin{align*}
(5) \quad A^t & := \lambda A^{t-1} + (1 - \lambda) P_s ; \\
(6) \quad u^2_t & := \lambda u + (1 - \lambda) ; \\
(7) \quad x^t_j & := \lambda x^t_j \text{ for } j \neq s \text{ and } x^t_s = \lambda x^t_{s-1} + (1 - \lambda) ;
\end{align*}

where the $\lambda$ which minimizes $u_t = \|A^t\|$ is:

$$\lambda := \frac{(1 - \nu)}{(u^2_{t-1} - 2\nu + 1)} ;$$

Note that $0 < \lambda < 1$ because $\nu = (A^{t-1})^T P_s \leq 0$. Also note that $u_t < u_{t-1}$ because in the right triangle $0A^{t-1}A^t$ the hypotenuse is $u_{t-1} = 0A^{t-1}$ and a leg is $u_t = 0A^t$.

Step 4: Return to Step 1 with $t := t + 1$.

Rate of Convergence

In the figure, $u_{t-1} = \|A^{t-1}\|$, $u_t = \|A_t\|$ and $1 = \|P_s\|$. Let the acute angle at $A^{t-1}$ be $\theta_1$, the acute angle at $P_s$ be $\theta_2$, and the acute exterior angle of the triangle at the origin be $\phi$. We have

$$\theta_1 + \theta_2 = \phi \leq \pi/2$$

where $\phi \leq \pi/2$ because, see Step 2, $\nu = (A^{t-1})^T P_s \leq 0$. Note that

$$\sin \theta_2 = \sin(\phi - \theta_1) \leq \sin(\pi/2 - \theta_1) = \cos \theta_1.$$

In the figure,

$$u_t/u_{t-1} = \sin \theta_1, \quad u_t/1 = \sin \theta_2 \leq \cos \theta_1, \quad (u_t/u_{t-1})^2 + (u_t/1)^2 \leq \sin^2 \theta_1 + \cos^2 \theta_1 = 1.$$
Dividing by $u_t^2$, it follows inductively that
\[
\frac{1}{u_{t-1}^2} + 1 \leq \frac{1}{u_t^2} \\
\frac{1}{u_{t-2}^2} + 1 \leq \frac{1}{u_{t-1}^2} \\
\vdots \quad \vdots \quad \vdots \\
\frac{1}{u_1^2} + 1 \leq \frac{1}{u_2^2} \quad \text{where } u_t = \|A^t\| = 1.
\]

We sum these $t-1$ inequalities to obtain $t \leq (1/u_t)^2 = 1/\|A^t\|^2$.

We conclude that if we wish to iterate until $u_t = \|A^t\| = \epsilon$, it can be attained in less than $1/\epsilon^2$ iterates. This result is independent of the dimensions $m$ and $n$. The number of arithmetic operations per iteration of the various steps:

<table>
<thead>
<tr>
<th></th>
<th>Multiplications</th>
<th>Additions</th>
<th>Comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>$\delta mn$</td>
<td>$\delta mn$</td>
<td>$n$</td>
</tr>
<tr>
<td>Step 3</td>
<td>$2m + n + 9$</td>
<td>$m + 8$</td>
<td>$1$</td>
</tr>
</tbody>
</table>

Complexity of von Neumann vs Karmarkar Algorithms

To attain a precision of $\epsilon$, the Karmarkar Algorithm has an upper bound of $n(- \log_\epsilon \epsilon)/\gamma$ iterations where $\gamma = 1 - \log_2 2 = 0.3$. The work per iteration for his algorithm is considerably higher than a von Neumann iteration. To load the dice in favor of the Karmarkar algorithm, we will assume the work per iteration is the same. The comparison is therefore for the same $\epsilon$ precision:

**Karmarkar:**
- Work $< (1/3)(- \log_\epsilon \epsilon)n(2\delta mn + 3m + 2n + 18)$ operations

**Von Neumann:**
- Work $< (1/\epsilon^2)(2\delta mn + 3m + 2n + 18)$ operations

For a precision $\epsilon$, say $\epsilon = 10^{-10}$, the von Neumann method has an upper
bound on the number of arithmetic operations which is lower than that for Karmarkar when the number of variables

\[ n > .3(- \log_e \epsilon) / \epsilon^2 = 1.3 \times 10^{18} \text{ if } \epsilon = 10^{-10}. \]

The polynomial complexity of Karmarkar as simplified above is 3 (actually it is 3.5) with a very small constant factor \((- \log_e \epsilon) / \gamma \simeq 75\) for \(\epsilon = 10^{-10}\), while von Neumann has a polynomial complexity of 2 but a constant factor of \(1/\epsilon^2 = 10^{20}\) to obtain a precision of \(10^{-10}\).

Making the von Neumann Algorithm more efficient

In tests on small examples, the convergence of the von Neumann algorithm was observed to be too slow for practical problems. In a companion paper to this one, a variant of the algorithm is presented that yields under certain conditions an exact solution in considerably fewer iterations, [2]. It too may be too slow to be practical, but exploits a "bracketing" idea which one may wish to explore as a way to speed up other infinitely converging methods. In this companion paper, instead of locating the point 0 at the center of the hypersphere as in the von Neumann algorithm, we locate the point off center at \(Q\). This change requires a small alteration in the proof of convergence: \((u_t/1) = \sin \theta_2 \leq \cos \theta_1\) is changed to read

\[(u_t/\|QP_t\|) = \sin \theta_2 \leq \cos \theta_1\]

where now \(\|QP_t\|^2 \leq 1 + \|Q\|^2 \leq 2\) instead of \(\|OP_1\| = 1\). This results in

\[ (1/u_{r-1})^2 + 1/(1 + \|Q\|)^2 \leq (1/u_r)^2 \quad \text{for } \tau = 2, \ldots, t \]

Summing over \(\tau = 2, \ldots, t\) yields the new upper bound \(t - 1 \leq (1 + \|Q\|)^2 / \epsilon^2\).
References


Abstract

Von Neumann, in a private communication to the author in 1948, proposed the first interior algorithm for finding a feasible solution to a linear program with a convexity constraint. I prove it has the remarkable property that, independent of the number of rows m and columns n, it generates in less than $1/e^2$ iterations a feasible solution with a precision $\varepsilon$ (where $e^2$ is the sum of the squares of errors of fit of the left hand side of the equations to the right hand side) when the general problem is recast in the form:

$$x \geq 0, \quad \sum_{i} x_i = 1, \quad \sum_{i} P_{ij} x_i = 0, \quad ||P||_2 = 1 \quad \forall j.$$

Letting $\delta$, $0 < \delta \leq 1$, denote the non-zero coefficient density, the work per iteration is $5mn + 2m + n + 9$ multiplications and $5mn + m + n + 9$ additions or comparisons.