A Comparison Among Simple Algorithms for Linear Programming

Jair da Silva
Federal University of Paraná-UFPR
Aurelio R.L. Oliveira, Carla T. L. S. Ghidini
University of Campinas-UNICAMP
Marta I. Velazco
Campo Limpo Paulista School

Funded by CNPq/FAPESP

Problem

Consider the search for a feasible solution for the following set of linear restrictions:

$$\begin{aligned} Px &= 0, \\ e^t x &= 1, \\ x &\geq 0, \end{aligned} \tag{1}$$

where $P \in \Re^{m \times n}$, $x \in \Re^n$ and $e \in \Re^n$ and be the vector with all the coordinates equal to one and the P columns have norm one, i.e., $||Pj|| = 1$, for $j = 1, \dots, n$.

Note that any linear programming problem can be reduced to the problem (1).

von Neumann's Algorithm

Given: $x^0 \geq 0$, with $e^t x^0 = 1$. Compute $b^0 = Px^0$.

For $k = 1, 2, 3, \ldots$ **Do:**

1) Compute:

$$s = \text{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\},$$
$$v^{k-1} = P_s^t b^{k-1}.$$

2) If $v^{k-1} > 0$, then **STOP**. The problem (1) is infeasible.

3) Compute:

$$u^{k-1} = ||b^{k-1}||,$$
$$\lambda = \frac{1-v^{k-1}}{(u^{k-1})^2 - 2v^{k-1} + 1}.$$

4) Update:

$$b^k = \lambda b^{k-1} + (1-\lambda)P_s,$$
$$x^k = \lambda x^{k-1} + (1-\lambda)e_s,$$

End

von Neumann's Algorithm

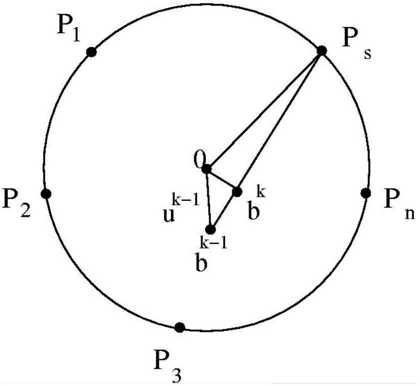


Illustration of the von Neumann's Algorithm.

Optimal Pair Adjustment Algorithm

Given: $x^0 \geq 0$, with $e^t x^0 = 1$. Compute $b^0 = Px^0$.

For $k = 1, 2, 3, \ldots$ **Do:**

1) Compute:

$$s^+ = \text{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\},$$
$$s^- = \text{argmax}_{j=1,\ldots,n}\{P_j^t b^{k-1} |\ x_j > 0\},$$
$$v^{k-1} = P_{s^+}^t b^{k-1}.$$

2) If $v^{k-1} > 0$, then **STOP**; the problem (1) is infeasible.

3) Solve the problem:

$$\begin{array}{ll} \text{minimize} & ||\overline{b}||^2 \\ \text{s.t.} & \lambda_0(1 - x_{s^+}^{k-1} - x_{s^-}^{k-1}) + \lambda_1 + \lambda_2 = 1, \\ & \lambda_i \geq 0, \ \text{for } i = 0, 1, 2. \end{array} \qquad (2)$$

where,

$$\overline{b} = \lambda_0(b^{k-1} - x_{s^+}^{k-1} P_{s^+} - x_{s^-}^{k-1} P_{s^-}) + \lambda_1 P_{s^+} + \lambda_2 P_{s^-}.$$

Optimal Pair Adjustment Algorithm

4) Update:

$$b^k = \lambda_0 (b^{k-1} - x_{s^+}^{k-1} P_{s^+} - x_{s^-}^{k-1} P_{s^-}) + \lambda_1 P_{s^+} + \lambda_2 P_{s^-},$$

$$x_j^k = \begin{cases} \lambda_0 x_j^{k-1}, & j \neq s^+ \text{ e } j \neq s^-, \\ \lambda_1, & j = s^+, \\ \lambda_2, & j = s^-. \end{cases}$$

End.

Optimal Adjustment Algorithms for p Coordinates

Given: $x^0 \geq 0$, with $e^t x^0 = 1$. Compute $b^0 = Px^0$.
For $k = 1, 2, 3, \ldots$ **Do:**
1) Compute:

$\{P_{\eta_1^+}, \ldots, P_{\eta_{s_1}^+}\}$ forming the largest angle with the vector b^{k-1}.
$\{P_{\eta_1^-}, \ldots, P_{\eta_{s_2}^-}\}$ forming the smallest angle with the vector b^{k-1}
such as $x_i^{k-1} > 0, i = \eta_1^-, \ldots, \eta_{s_2}^-$, where $s_1 + s_2 = p$.

$v^{k-1} = \text{minimum}_{i=1,\ldots,s_1}\{P^t_{\eta_i^+} b^{k-1}\}$.
2) If $v^{k-1} > 0$, then **STOP**; the problem (1) is infeasible.

Optimal Adjustment Algorithms for p Coordinates

3) Solve the problem:

$$\begin{array}{ll} \text{minimize} & ||\overline{b}||^2 \\ \text{s.t.} & \lambda_0 \left(1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1}\right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} = 1, \\ & \lambda_{\eta_i^+} \geq 0, \quad \text{for } i = 1, \ldots, s_1, \\ & \lambda_{\eta_j^-} \geq 0, \quad \text{for } j = 1, \ldots, s_2, \end{array} \tag{3}$$

where

$$\overline{b} = \lambda_0 \left(b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} P_{\eta_j^-}\right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} P_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} P_{\eta_j^-}.$$

Optimal Adjustment Algorithms for p Coordinates

4) Update:

$$b^k = \lambda_0 \left(b^{k-1} - \sum_{i=1}^{s_1} x^{k-1}_{\eta_i^+} P_{\eta_i^+} - \sum_{j=1}^{s_2} x^{k-1}_{\eta_j^-} P_{\eta_j^-} \right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} P_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} P_{\eta_j^-},$$

$$x_j^k = \begin{cases} \lambda_0 x_j^{k-1}, & j \notin \{\eta_1^+, \ldots, \eta_{s_1}^+, \eta_1^-, \ldots, \eta_{s_2}^-\}, \\ \lambda_{\eta_i^+}, & j = \eta_i^+, i = 1, \ldots, s_1, \\ \lambda_{\eta_j^-}, & j = \eta_j^-, j = 1, \ldots, s_2. \end{cases}$$

End

- **We choose the column that forms the largest angle with the residue** b^k**.**

- We choose the column that forms the largest angle with the residue b^k.
- **Thus, the Steps** 1) **and** 2) **of algorithms are exactly the same.**

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- **von Neumann's Algorithm**

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) **Compute:**

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \textbf{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\}$,

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \text{argmin}_{j=1,\dots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_s^t b^{k-1}$.

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \text{argmin}_{j=1,\dots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_s^t b^{k-1}$.
- 2) **If** $v^{k-1} > 0$, **then STOP. The problem (1) is infeasible.**

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \text{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_s^t b^{k-1}$.
- 2) If $v^{k-1} > 0$, then **STOP**. The problem (1) is infeasible.
- **Algorithm with p=1**

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \text{argmin}_{j=1,\dots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_s^t b^{k-1}$.
- 2) If $v^{k-1} > 0$, then **STOP**. The problem (1) is infeasible.
- Algorithm with p=1
- 1) **Compute:**

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \text{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_s^t b^{k-1}$.
- 2) If $v^{k-1} > 0$, then **STOP**. The problem (1) is infeasible.
- Algorithm with p=1
- 1) Compute:
- $s^+ = \textbf{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\}$,

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \text{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_s^t b^{k-1}$.
- 2) If $v^{k-1} > 0$, then **STOP**. The problem (1) is infeasible.
- Algorithm with p=1
- 1) Compute:
- $s^+ = \text{argmin}_{j=1,\ldots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_{s^+}^t b^{k-1}$.

- We choose the column that forms the largest angle with the residue b^k.
- Thus, the Steps 1) and 2) of algorithms are exactly the same.
- von Neumann's Algorithm
- 1) Compute:
- $s = \text{argmin}_{j=1,\dots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_s^t b^{k-1}$.
- 2) If $v^{k-1} > 0$, then **STOP**. The problem (1) is infeasible.
- Algorithm with p=1
- 1) Compute:
- $s^+ = \text{argmin}_{j=1,\dots,n}\{P_j^t b^{k-1}\}$,
- $v^{k-1} = P_{s^+}^t b^{k-1}$.
- 2) **If** $v^{k-1} > 0$, **then STOP; the problem (1) is infeasible.**

Equivalence between algorithms with $p = 1$ and von Neumann

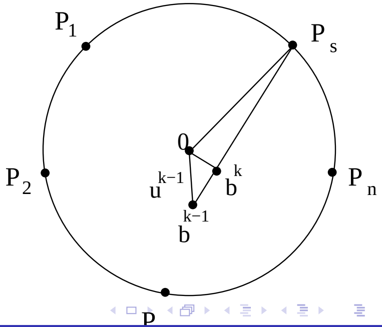
- **They differ in the Step 3). In the von Neumann's algorithm.**

We solve the following subproblem

$$\begin{array}{ll} \text{minimize} & ||\overline{b}||^2 \\ \text{s.t.} & \lambda \in [0, 1] \end{array}$$

where, $\overline{b} = \lambda b^{k-1} + (1 - \lambda) P_s$.

This is done by computing the optimal λ.



Equivalence between algorithms with $p = 1$ and von Neumann

In the algorithm with $p = 1$,

we have to solve the subproblem.

$$\begin{aligned} &\text{minimize} \quad ||\overline{b}||^2 \\ &\text{s.t.} \quad \lambda_0(1 - x_s^{k-1}) + \lambda_1 = 1, \\ &\lambda_i \geq 0, \text{ for } i = 0, 1. \end{aligned}$$

where,

$$\overline{b} = \lambda_0(b^{k-1} - x_s^{k-1} P_s) + \lambda_1 P_s.$$

We can rewrite the subproblem as follows:

$$\lambda_0(1 - x_s^{k-1}) + \lambda_1 = 1 \Leftrightarrow$$

$$\lambda_1 = 1 - \lambda_0(1 - x_s^{k-1}) \geq 0$$

$$\begin{aligned} \overline{b} &= \lambda_0(b^{k-1} - x_s^{k-1} P_s) + \lambda_1 P_s \\ &= \lambda_0(b^{k-1} - x_s^{k-1} P_s) + \\ &\quad (1 - \lambda_0(1 - x_s^{k-1})) P_s \\ &= \lambda_0 b^{k-1} + (1 - \lambda_0) P_s. \end{aligned}$$

and

$$0 \leq \lambda_0 \leq \frac{1}{1 - x_s^{k-1}}$$

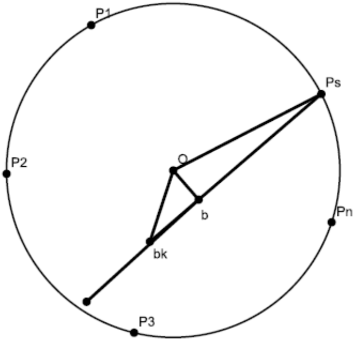
Equivalence between algorithms with $p = 1$ and von Neumann

This, we have the problem

$$\begin{array}{ll} \text{minimize} & ||\overline{b}||^2 \\ \text{s.t.} & \lambda \in \left[0, \frac{1}{1-x_s^{k-1}}\right] \end{array}$$

where, $\overline{b} = \lambda b^{k-1} + (1-\lambda)P_s.$

As $\frac{1}{1-x_s^{k-1}} > 1$, so the geometric view of algorithm with $p = 1$ is given by Figure 4.



Equivalence between algorithms with $p = 1$ and von Neumann

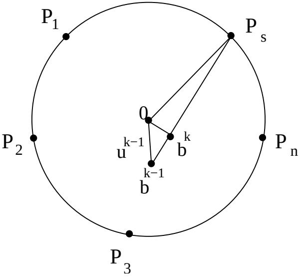


Figura: Illustration of the von Neumann's algorithm

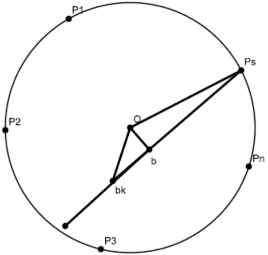


Figura: Illustration of the algorithm with $p = 1$.

Equivalence between algorithms with $p = 1$ and von Neumann

In the Step 4) of the von Neumann's algorithm x^k is updated by

$$x_j^k = \begin{cases} \lambda x_j^{k-1}, & j \neq s \\ \lambda(x_s^{k-1} - 1) + 1, & j = s. \end{cases}$$

In the Step 4) of the algorithm with $p = 1$, x^k is updated by

$$x_j^k = \begin{cases} \lambda x_j^{k-1}, & j \neq s \\ 1 - \lambda(1 - x_s^{k-1}), & j = s. \end{cases}$$

Geometric view of the Optimal Adjustment Algorithms for p Coordinates

For the case $p = 2$, the subproblem of Step 3) reduces to the following form:

$$\begin{array}{ll} \text{minimize} & ||\overline{b}||^2 \\ \text{s.t.} & \lambda_0(1 - x_{s^+}^{k-1} - x_{s^-}^{k-1}) + \lambda_1 + \lambda_2 = 1, \quad \text{where,} \\ & \lambda_i \geq 0, \text{ for } i = 0, 1, 2. \end{array}$$

$$\overline{b} = \lambda_0(b^{k-1} - x_{s^+}^{k-1} P_{s^+} - x_{s^-}^{k-1} P_{s^-}) + \lambda_1 P_{s^+} + \lambda_2 P_{s^-}.$$

We can rewrite $\overline{b}$ as:

$$\begin{aligned} \overline{b} &= \lambda_0(b^{k-1} - x_{s^+}^{k-1} P_{s^+} - x_{s^-}^{k-1} P_{s^-}) + \lambda_1 P_{s^+} + \lambda_2 P_{s^-} \\ &= \lambda_0 b^{k-1} + (\lambda_1 - \lambda_0 x_{s^+}^{k-1}) P_{s^+} + (\lambda_2 - \lambda_0 x_{s^-}^{k-1}) P_{s^-} \end{aligned}$$

and as we have $\lambda_0 + (\lambda_1 - \lambda_0 x_{s^+}^{k-1}) + (\lambda_2 - \lambda_0 x_{s^-}^{k-1}) = 1$, then $\overline{b}$ is an affine combination.

Geometric view of the Optimal Adjustment Algorithms for p Coordinates

- **The set formed by all combinations is the plane containing the points b^{k-1}, P_{s^+} and P_{s^-}.**

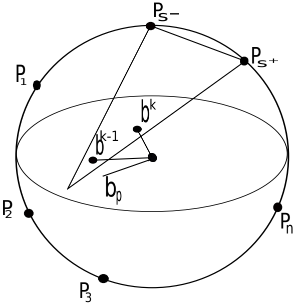


Figura: Illustration of the algorithm with $p = 2$.

Geometric view of the Optimal Adjustment Algorithms for p Coordinates

- The set formed by all combinations is the plane containing the points b^{k-1}, P_{s^+} and P_{s^-}.
- $\overline{b}(\lambda_0, \lambda_1, \lambda_2)$ **is a linear transformation with domain formed by all points of the restriction** $\lambda_0(1-x_{s^+}^{k-1}-x_{s^-}^{k-1})+\lambda_1+\lambda_2 = 1$

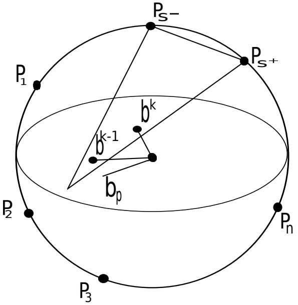


Figura: Illustration of the algorithm with $p = 2$.

Geometric view of the Optimal Adjustment Algorithms for p Coordinates

- **The image this set by linear transformation is the triangle with vertices in P_{s^+}, P_{s^-} and P_v and its interior in the affine space, where** $P_v = \frac{1}{(1-x_{s^+}^{k-1}-x_{s^-}^{k-1})}(b^{k-1} - x_{s^+}^{k-1}P_{s^+} - x_{s^-}^{k-1}P_{s^-})$

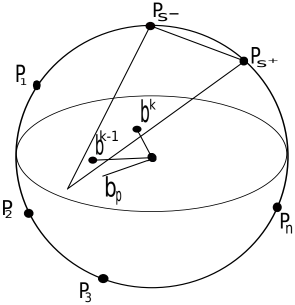


Figura: Illustration of the algorithm with $p = 2$.

Geometric view of the Optimal Adjustment Algorithms for p Coordinates

- The image this set by linear transformation is the triangle with vertices in P_{s^+}, P_{s^-} and P_v and its interior in the affine space, where $P_v = \frac{1}{(1-x_{s^+}^{k-1}-x_{s^-}^{k-1})}(b^{k-1} - x_{s^+}^{k-1}P_{s^+} - x_{s^-}^{k-1}P_{s^-})$
- **The optimal residue b^k that we seek is the projection of the origin on this triangle**

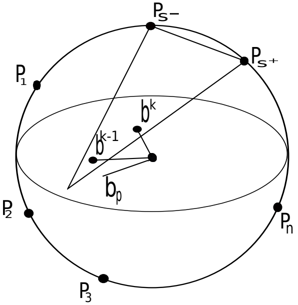


Figura: Illustration of the algorithm with $p = 2$.

Geometric view of the Optimal Adjustment Algorithms for p Coordinates

- **In the case of the algorithm for p coordinates**

Geometric view of the Optimal Adjustment Algorithms for p Coordinates

- In the case of the algorithm for p coordinates
- **The optimal residue b^k is the projection of the origin on the affine space region with vertices in the p columns and in the vector P_v.**

Subproblem Solution Using Interior Points Methods

First, we rewrite the sub-problem in the matrix format:

$$\begin{aligned} &\text{minimize } \tfrac{1}{2}||W\lambda||^2 \\ &\text{s.a } \; c^t\lambda = 1, \\ &\qquad -\lambda \leq 0, \end{aligned} \tag{4}$$

$$\begin{aligned} W &= \left[\overline{w}\; P_{\eta_1^+} \dots P_{\eta_{s_1}^+} P_{\eta_1^-} \dots P_{\eta_{s_2}^-}\right], \\ \overline{w} &= b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} P_{\eta_j^-}, \\ \lambda &= \left(\lambda_0, \lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \dots, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}\right), \\ c &= (c_1, 1, \dots, 1), \; c_1 = 1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1}. \end{aligned} \tag{5}$$

Subproblem Solution Using I nterior Points Methods

The KKT equations from the problem (4) are given by:

$$\begin{aligned} W^tW\lambda + cl - \mu &= 0 \\ \mu^t\lambda &= 0 \\ c^t\lambda - 1 &= 0, \end{aligned} \qquad (6)$$

where τ is free, $0 \leq \mu$ are the Lagrange multipliers for equality and inequality, respectively, and the W^tW matrix is of the order $(p+1) \times (p+1)$.
Next, we apply the path-following interior point method to problem (6).

Subproblem Solution Using Interior Points Methods

The linear system that arises at the iteration of the interior point method applied to these equations has the form:

$$\begin{pmatrix} W^tW & c & -Id \\ U & 0 & \Lambda \\ c^t & 0 & 0 \end{pmatrix} \begin{bmatrix} d\lambda \\ dl \\ d\mu \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad (7)$$

where

$$\begin{aligned} U &= diag(\mu), \\ \Lambda &= diag(\lambda), \\ r_1 &= \mu - cl - W^tW\lambda, \\ r_2 &= -l^t\lambda, \\ r_3 &= 1 - c^t\lambda. \end{aligned}$$

Subproblem Solution Using Interior Points Methods

$$d\mu = \Lambda^{-1} r_2 - \Lambda^{-1} U d\lambda,$$
$$d\lambda = (W^t W + \Lambda^{-1} U)^{-1} r_4 - (W^t W + \Lambda^{-1} U)^{-1} c\, dl,$$
$$c^t (W^t W + \Lambda^{-1} U)^{-1} c\, dl = c^t (W^t W + \Lambda^{-1} U)^{-1} r_4 - r_3,$$

$$(W^t W + \Lambda^{-1} U) sl1 = c$$
$$(W^t W + \Lambda^{-1} U) sl2 = r_4$$
$$r_4 = r_1 + \Lambda^{-1} r_2$$

Theoretical Properties of the New Method

Theorem

The decrease in $||b^k||$ obtained by an iteration of the optimal adjustment algorithm for p coordinates, with $1 \leq p \leq n$, where n is the number of P columns, in the worst scenario, is equal to the one obtained by an iteration of the von Neumann's algorithm

Theoretical Properties of the New Method

Theorem

The decrease in $||b^k||$ obtained by an iteration of the optimal adjustment algorithm for p_2 coordinates, in the worst scenario, is equal to the one obtained by an iteration of the optimal adjustment algorithm for p_1 coordinates with $p_1 \leq p_2 \leq n$, where n is the number of columns P.

Sufficient Condition for $||b^k|| < ||b^k_v||$

- Let b^k be the residue of the algorithm with $p = 2$ in the iteration k and let P_{s^+} and P_{s^-} be the columns forming the largest and smallest angles with the vector b^{k-1}.

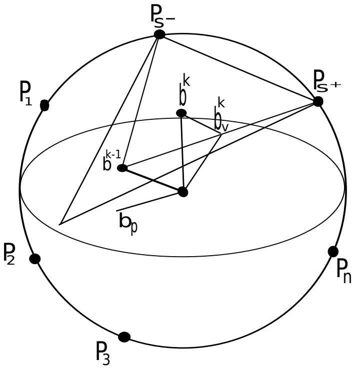


Figura: Illustration of the Sufficient Condition.

Sufficient Condition for $||b^k|| < ||b_v^k||$

- Let b^k be the residue of the algorithm with $p = 2$ in the iteration k and let P_{s^+} and P_{s^-} be the columns forming the largest and smallest angles with the vector b^{k-1}.
- If the projection of the origin is in the interior of the triangle $b^k P_{s^+} P_{s^-}$

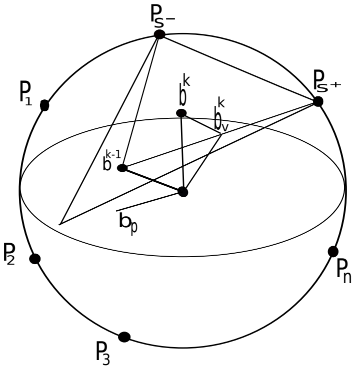


Figura: Illustration of the Sufficient Condition.

Sufficient Condition for $||b^k|| < ||b_v^k||$

- **And coincides with the projection of the origin in the plane determined by b^{k-1}, P_{s^+} and P_{s^-}, then $||b^k|| < ||b_v^k||$, where b_v^k is the residue of the von Neumann's algorithm in the iteration k.**

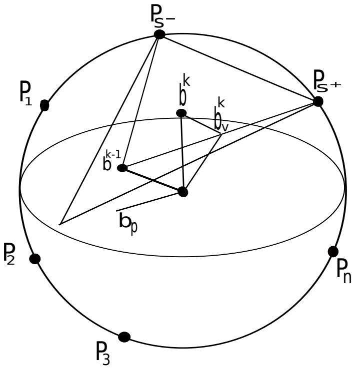


Figura: Illustration of the Sufficient Condition.

Sufficient Condition for $||b^k|| < ||b_v^k||$

- And coincides with the projection of the origin in the plane determined by b^{k-1}, P_{s^+} and P_{s^-}, then $||b^k|| < ||b_v^k||$, where b_v^k is the residue of the von Neumann's algorithm in the iteration k.
- **We can see in the Figure 8 that the triangle $0b^k b_v^k$ has the $\overline{0b_v^k}$ hypotenuse and side $\overline{0b^k}$.**

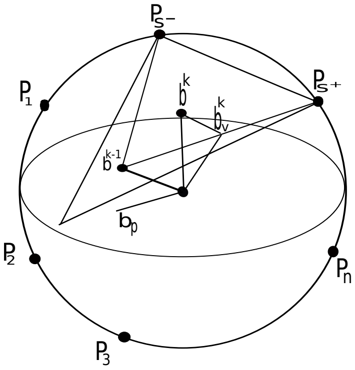


Figura: Illustration of the Sufficient Condition.

Computational Experiments

- **The main objective of our experiments was to analyze the performance of the family of algorithms for various and moderate value of p, with the optimal pair adjustment algorithm, with is the case when $p = 2$.**

Computational Experiments

- The main objective of our experiments was to analyze the performance of the family of algorithms for various and moderate value of p, with the optimal pair adjustment algorithm, with is the case when $p = 2$.
- **We used a collection of 151 linear programming problems.**

Computational Experiments

- The main objective of our experiments was to analyze the performance of the family of algorithms for various and moderate value of p, with the optimal pair adjustment algorithm, with is the case when $p = 2$.
- We used a collection of 151 linear programming problems.
- **The problems are divided into 95 Netlib problems, 16 Kennington problems, and 40 other problems, which are not publicly available, and were supplied by Gonçalves [5]**

Computational Experiments

- **The family of algorithms was implemented in C.**

Computational Experiments

- The family of algorithms was implemented in C.
- **To solve the subproblem, the classical path-following interior point method was implemented in C**

Computational Experiments

1. **First, the von Neumann's algorithm is ran on all problems;**

Computational Experiments

1. First, the von Neumann's algorithm is ran on all problems;
2. **Then, when the relative difference between $||b^{k-1}||$ and $||b^k||$ was less than 0.5%, the time $t1$(CPU seconds) and number of iteration (up to $t1$) are recorded .**

Computational Experiments

1. First, the von Neumann's algorithm is ran on all problems;
2. Then, when the relative difference between $||b^{k-1}||$ and $||b^k||$ was less than 0.5%, the time $t1$(CPU seconds) and number of iteration (up to $t1$) are recorded .
3. **Next, the times $t2$, $t3$, $t4$ and $t5$ (CPU seconds), which respectively correspond to 3, 5, 10 and 20 times the number of iterations in $t1$ are also recorded.**

Computational Experiments

1. First, the von Neumann's algorithm is ran on all problems;
2. Then, when the relative difference between $||b^{k-1}||$ and $||b^k||$ was less than 0.5%, the time $t1$(CPU seconds) and number of iteration (up to $t1$) are recorded .
3. Next, the times $t2$, $t3$, $t4$ and $t5$ (CPU seconds), which respectively correspond to 3, 5, 10 and 20 times the number of iterations in $t1$ are also recorded.
4. **After that, the optimal adjustment algorithm for p coordinates where $p = 2$, $p = 4$, $p = 10$ and $p = 20$ is ran on the test problems.**

Computational Experiments

1. First, the von Neumann's algorithm is ran on all problems;
2. Then, when the relative difference between $||b^{k-1}||$ and $||b^k||$ was less than 0.5%, the time $t1$(CPU seconds) and number of iteration (up to $t1$) are recorded .
3. Next, the times $t2$, $t3$, $t4$ and $t5$ (CPU seconds), which respectively correspond to 3, 5, 10 and 20 times the number of iterations in $t1$ are also recorded.
4. After that, the optimal adjustment algorithm for p coordinates where $p = 2$, $p = 4$, $p = 10$ and $p = 20$ is ran on the test problems.
5. **Finally, for the ti times, $i = 1, \ldots, 5$, the residue $||b^k||$ is recorded.**

Computational Results

Tabela: Percentage of the relative gain by the algorithms on the problems in five different times

Algorithm	t1	t2	t3	t4	t5
Algorithm with p=2	15,89%	11,92%	22,51%	5,96%	7,94 %
Algorithm with p=4	31,12%	34,43%	27,81%	19,96%	23,83 %
Algorithm with p=10	20,52%	27,15%	25,16%	25,16%	23,84 %
Algorithm with p=20	32,45%	26,49%	24,50%	50,99%	44,37%

Computational Results

- We analyze the performances of the algorithms using performance profile.

Computational Results

- We analyze the performances of the algorithms using performance profile.
- **The distance of the residue $||b^k||$ to the origin, was used to measure the performance.**

Computational Results

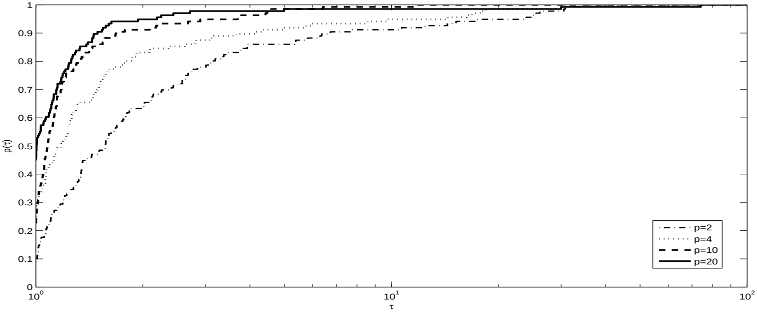


Figura: Profile Performance of four algorithms in t5 time.

Computational Results

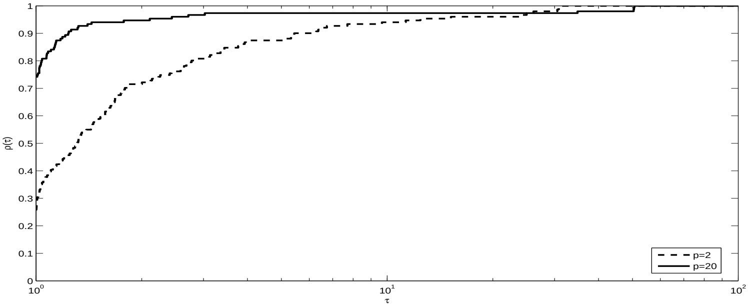


Figura: Performance profile of the algorithms with $p = 2$ and $p = 20$ in time $t5$.

Computational Results

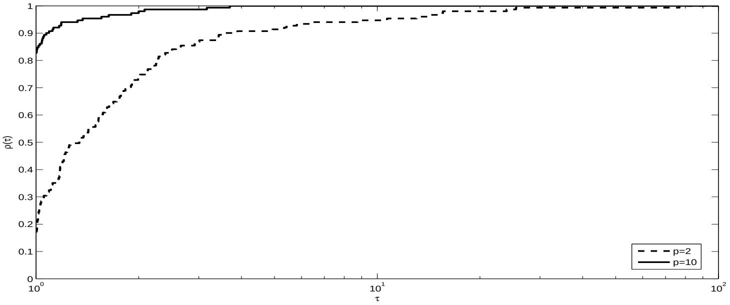


Figura: Performance profile of the algorithms with $p = 2$ and $p = 10$ in time $t5$.

Computational Results

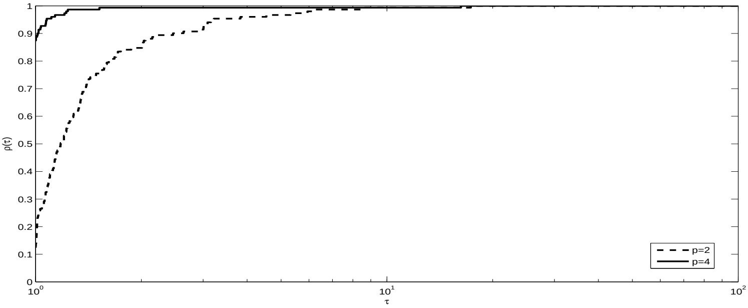


Figura: Performance profile of the algorithms with $p = 2$ and $p = 4$ in time $t5$.

Thank You

G. B. Dantzig, Converting a converging algorithm into a polynomially bounded algorithm, Technical Report SOL 91-5, Stanford University, 1991.

G. B. Dantzig, An ϵ-precise feasible solution to a linear program with a convexity constraint in $\frac{1}{\epsilon^2}$ iterations independent of problem size, Technical Report SOL 92-5, Stanford University, 1992.

M. Epelman and R. M. Freund, Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system, *Mathematical Programing*, **88**, (2000) 451-485.

J.P.M. Gonçalves, "A Family of Linear Programming Algorithms Based on the Von Neumann Algorithm", PhD thesis, Lehigh University, Bethlehem, PA, 2004.

- J.P.M. Gonçalves, R.H. Storer and J.Gondzio. New Algorithms for Linear Programming Based on an Algorithm by Von Neumann, Technical Report, Lehigh University, 2005.
- S. Bocanegra, F. F. Campos, A. R. L. Oliveira. Using a Hybrid Preconditioner for Solving Large-Scale Linear Systems arising from Interior Point Methods, *Computational Optimization and Applications*, Vol 36, pp. 149-164, 2007.