

A new algorithm for solving linear programming problems

Un nuevo algoritmo para la solución de problemas de programación lineal

A. L. Ramírez¹, O. Buitrago², R. A. Britto³, A. Fedossova⁴

ABSTRACT

Linear programming (LP) is one of the most widely-applied techniques in operations research. Many methods have been developed and several others are being proposed for solving LP problems, including the famous simplex method and interior point algorithms. This study was aimed at introducing a new method for solving LP problems. The proposed algorithm starts from an interior point and then carries out orthogonal projections using parametric straight lines to move between the interior and polyhedron frontier defining the feasible region until reaching the extreme optimal point.

Keywords: linear programming, optimisation, orthogonal projection, parametric equation.

RESUMEN

La programación lineal (PL) es una de las herramientas de mayor aplicación en la investigación de operaciones. Se han desarrollado y se siguen proponiendo varios métodos para la resolución de problemas de este tipo, desde el famoso simplex hasta los algoritmos de punto interior. Este trabajo tiene como propósito principal presentar la propuesta de un nuevo procedimiento para la solución de problemas PL que, partiendo de un punto interior, realiza proyecciones ortogonales mediante rectas paramétricas y se mueve iterativamente entre el interior y la frontera del poliedro que define la región factible hasta llegar al punto extremo óptimo.

Palabras clave: programación lineal, optimización, proyecciones ortogonales, ecuaciones paramétricas.

Received: June 21th 2011

Accepted: May 20th 2012

Introduction

Linear programming (LP) dates from 1939 when Leonid Kantorovich first expressed a problem in economics in linear form (Bazaraa *et al.*, 1998). However, LP as an area of study has been attributed to George Dantzig who also developed the simplex method in 1947 (Cottle, 2006). LP has been widely studied in engineering programmes, management, mathematics and other fields due to its applicability.

Since its publication, the simplex method (and its variants) was the main tool for solving linear problems until 1980 when Khachiyan proposed an ellipsoidal algorithm (Khachiyan, 1982). Although its practical implementation was not very efficient, nevertheless it served as the basis for designing Karmarkar's interior point algorithm (IPA), which is computationally efficient for solving large-sized problems (Karmarkar, 1984) and computationally less complex than the simplex method (Powell, 1993);

many additional contributions have been made since then, for example, a primal-dual simplex algorithm (Norman and Curet, 1993) which uses less basis than the original simplex primal-dual, another solves the primal-dual starting from an infeasible interior point (Mizuno, 1994), a projective simplex algorithm using LU decomposition (Pan, 2000), an extension of Karmarkar's IPA having higher convergence speed (Naseri and Valinejad, 2007) and an adaptive primal-dual IPA (Kim, Lee and Cho, 2009). More recently, a proposed IPA based on a new search direction (Zhang and Xu, 2011) and another method based on a new barrier function (Cho, 2011) have been developed.

Considering the large number of areas where LP can be applied and the importance of having efficient algorithms for solving LP problems, any contribution made to LP is very relevant. Consequently, a new method for solving LP problems is presented in this paper.

Formulating the problem

An LP problem's general structure is well-known:

$z = d_1x_1 + d_2x_2 + d_3x_3 + \dots + d_jx_j + d_{j+1}x_{j+1} + \dots + d_nx_n$ is maximized (minimised) subject to:

$$\begin{array}{cccccccc} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 & + \dots + a_{1j}x_j & + \dots + a_{1n}x_n & \leq & b_1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 & + \dots + a_{ij}x_j & + \dots + a_{in}x_n & \leq & b_i \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 & + \dots + a_{mj}x_j & + \dots + a_{mn}x_n & \leq & b_m \\ & & & & x_1, x_2, x_3, \dots, x_j, \dots, x_n & \geq 0 \end{array}$$

¹ Andrés Leonardo Ramírez Leal. Affiliation: Universidad de La Salle. MSc in mathematics, Universidad Nacional de Colombia. BSc. in Chemistry Universidad Nacional de Colombia. E-mail: andres1729@yahoo.com

² Oscar Yecid Buitrago Suescún. Affiliation: Universidad Militar Nueva Granada, Colombia. MSc in Industrial engineering, Universidad de los Andes, Colombia. BSc. in Chemical Engineering, Universidad Nacional de Colombia. E-mail: os-car.buitrago@unimilitar.edu.co

³ Rodrigo Alberto Britto Agudelo. Affiliation: Universidad de Los Andes, Colombia. PHD in Logistics, Maryland University, United States. MSc in Industrial engineering, Universidad de los Andes, Colombia. BSc. in Chemical engineering, Universidad Nacional de Colombia. E-mail: rab@adm.uniandes.edu.co

⁴ Alina Fedossova. Affiliation: Universidad Nacional de Colombia. PhD in Mathematics, Moscú State University Lomonosov. BSc. in Mathematics, Moscú State University Lomonosov. E-mail: afedosova@unal.edu.co

How to cite: Ramírez A. L., Buitrago O., R. Britto A., Fedossova A. (2012). A new algorithm for solving linear programming problems. Ingeniería e Investigación. Vol. 32, No. 2, August 2012, pp. 68-73.

d_j being the coefficients of the objective function, x_j decision variables, a_{ij} the coefficient of the j -th variable in the i -th constraint and b_i the right side of the i -th constraint.

An LP problem can also be written in a compact form as follows:

$$\max (\min) z = \mathbf{d}\mathbf{x}, \text{ subject to : } \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \quad (1)$$

where \mathbf{d} denotes the objective function coefficients' vector, \mathbf{A} the matrix of constraint coefficients, \mathbf{x} a variable vector, \mathbf{b} the vector of variables and $\mathbf{0}$ a vector made up only of zeros.

A maximisation problem will be used from now on as the basis; non-negative constraints have not included since the algorithm did not require them (if required they could have been added as additional constraints). If f_j were to represent the j -th row of matrix \mathbf{A} in (1), the LP problem could have been rewritten as: $\max z = \mathbf{d}\mathbf{x}$, subject to: $f_j \mathbf{x} \leq b_j \quad \forall j = 1, 2, \dots, m$ and hyperplane $f_j \mathbf{x} = b_j$ would have been symbolised as π_j

Describing the proposed algorithm

The proposed algorithm had two components: an initialisation stage (non-iterative) and the main cycle (iterative).

Initialisation stage

$n-1$ orthogonal vectors to objective function gradient ($\nabla z = \mathbf{d}$) were defined. $n-1$ parametric straight lines from any interior point were also defined during this stage (see Winston, 2005, pp 597-604, explaining how to obtain an interior point with Karmarkar's algorithm). The $2(n-1)$ intersections of these straight lines were found with faces of a polyhedron defined by the problem's feasible region. The respective centroid C_0 was calculated, from which gradient \mathbf{d} was projected. This projection's intersection with one of the faces of the polyhedron ended the algorithm's initialisation. The steps involved in this stage were.

1. An interior point $\mathbf{P}_0 = (y_1, y_2, \dots, y_n) \in R^n$, was found, so that $\mathbf{A}\mathbf{P}_0 < \mathbf{b}$
2. $n-1$ orthogonal vectors $\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_1, \dots, \mathbf{O}_{n-1}$ to \mathbf{d} (∇z)
3. $n-1$ parametric straight lines $\mathbf{P}_0 + \mathbf{O}_i t, t \in R, 1 \leq i \leq n-1$ were defined.

$$\mathbf{P}_0 + \mathbf{O}_i t : \begin{cases} x_1 = y_1 + O_i^1 t \\ \vdots \\ x_n = y_n + O_i^n t \end{cases} \quad (2)$$

4. $n-1$ parametric straight lines' intersection with hyperplanes $\pi_j \quad 1 \leq j \leq m$ was found. For each i -th straight line there were no more than m intersections with hyperplanes π_j , defined by the vector of parameters $[t_{i1}, t_{i2}, \dots, t_{im}]$. This was calculated by substituting values of x_i according to (2) in each hyperplane equation:

$$a_{i1}(y_1 + O_i^1 t) + \dots + a_{in}(y_n + O_i^n t) = b_i$$

from where the value of t for straight line i intersection

with hyperplane π_j could be cleared:

$$t_{ij} = \frac{b_j - \mathbf{f}_j \cdot \mathbf{P}_0}{\mathbf{f}_j \cdot \mathbf{O}_i}$$

5. The least positive value of t and the least absolute value for the negative values of t were selected for the i -th straight line:

$$t_i^{pos} = \min_{t_j > 0} (t_{ij}) \quad \forall i \text{ and } t_i^{neg} = \min_{t_j < 0} (|t_{ij}|) \quad \forall i \quad (3)$$

6. Points of intersection $\mathbf{Q}_{i1} = \mathbf{P}_0 + \mathbf{O}_i t_i^{pos}$ and $\mathbf{Q}_{i2} = \mathbf{P}_0 + \mathbf{O}_i t_i^{neg}$ were calculated using the values of t in (3) for each straight line with the face of the polyhedron. Total $2(n-1)$ points were $\mathbf{Q}_{i1}, \mathbf{Q}_{i2}, \dots, \mathbf{Q}_{n-1,1}, \mathbf{Q}_{n-1,2}$
7. These points' centroid was:

$$\mathbf{C}_0 = \frac{\mathbf{Q}_{i1} + \mathbf{Q}_{i2} + \dots + \mathbf{Q}_{n-1,1} + \mathbf{Q}_{n-1,2}}{2(n-1)}$$

to construct straight line:

$$\mathbf{C}_0 + \mathbf{d} t : \begin{cases} x_1 = C_1 + d_1 t \\ \vdots \\ x_n = C_n + d_n t \end{cases} \quad (4)$$

and intersections with m hyperplanes were found. Substituting the values of obtained from (4) in the respective hyperplane equations, led to:

$$a_{i1}(C_1 + d_1 t) + \dots + a_{in}(C_n + d_n t) = b_j \quad \therefore \quad t_j = \frac{b_j - \mathbf{f}_j \cdot \mathbf{C}_0}{\mathbf{f}_j \cdot \mathbf{d}}$$

8. $t_{\min} = \min_{t_j > 0} (t_j)$ was defined. The non-existence of at least one positive value of t indicated that the problem was not bounded, thereby finishing the algorithm and reporting this condition.
9. Point \mathbf{F}_1 coordinates were calculated on the k -th face of the polyhedron:

$$\mathbf{F}_1 = \mathbf{C}_0 + \mathbf{d} t_{\min} \in \pi_k$$

Main cycle (iterative)

$n-1$ orthogonal vectors to objective function gradient ($\nabla z = \mathbf{d}$) were defined. $n-1$ parametric straight lines from any interior point were also defined during this stage

Once an algorithm has been initialised, iterative calculations must be made to find an optimal solution. Orthogonal projections were performed from \mathbf{F}_1 , and both its intersections on the faces of the polyhedron and the centroid of these intersections were obtained. From the centroid, the gradient of the objective function was projected and the intersection on one of the faces of the polyhedron was calculated. Repeating this procedure led to coming close to an optimal solution for the required level of precision. The steps in the iterative cycle were:

1. Finding $n-1$ orthogonal vectors $\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_1, \dots, \mathbf{O}_{n-1}$ to \mathbf{d} so that:

$$\text{sign}(\mathbf{F}_1 \mathbf{C}_{i-1} \cdot \mathbf{f}_k) = \text{sign}(\mathbf{O}_i \cdot \mathbf{f}_k) \text{ or zero}$$

where k was the index of the face on which the last projection was done.

2. Finding the intersections of each straight line $F_i + O_i t, t \in R$ with every m hyperplane (note that plane π_k could have been excluded):

$$F_i + O_i t: \begin{cases} x_1 = F_i + O_i^1 t \\ \vdots \\ x_n = F_i + O_i^n t \end{cases}$$

substituting and clearing led to: $t_{ij} = \frac{b_j - f_j \cdot F_i}{f_j \cdot O_i}$

3. Choosing minimum positive value of t for each straight line:

$$t_{i\min} = \min_{t_j > 0} (t_j) \quad \forall i$$

4. Calculating each straight line's (each minimum value of t) points of intersection with the faces of the polyhedron using:

$$Q_i = F_i + O_i t_{i\min}$$

5. Calculating the value of new centroid: $C_i = \frac{Q_1 + \dots + Q_{n-1}}{n-1}$

6. Constructing straight line $C_i + d t$ and finding its intersection with m hyperplanes π_j

$$C_i + d t: \begin{cases} x_1 = C_i + d_1 t \\ \vdots \\ x_n = C_i + d_n t \end{cases} \quad (5)$$

substituting values of x_i (5) in the in the respective hyperplane equations, led to:

$$a_{i1}(C_i + d_1 t) + \dots + a_{in}(C_i + d_n t) = b_i, \therefore t_j = \frac{b_j - f_j \cdot C_i}{f_j \cdot d}$$

7. Defining $t_{\min} = \min_{t_j > 0} (t_j)$. If the solution was unbounded, this condition was not necessarily detectable in the initialisation stage. This condition was still indicated by the non-existence of at least one positive value of t , in which case the algorithm had terminated.

8. Calculating the coordinates for point F_i on the k -th face of the polyhedron:

$$F_{i+1} = C_i + d t_{\min} \in \pi_k$$

9. Comparing the coordinates for F_{i+1} with F_i . This was finished if the established precision criterion was satisfied, otherwise one had to return to step 1 and perform another iteration.

Application example

A problem in R^2 is presented to illustrate each step in the algorithm and graphically visualise it.

The following problem was considered:

$$\begin{aligned} \text{Max } z &= 3x_1 + x_2 \\ \text{s.t. } & \frac{1}{2}x_1 + x_2 \leq 7; \quad \frac{3}{2}x_1 - x_2 \leq 12; \quad \frac{5}{4}x_1 + x_2 \leq 21 \\ & x_1 \geq 0; \quad x_2 \geq 0 \end{aligned}$$

Figure 1 shows the feasible region, the objective function gradient and the optimal solution ($x_1 = 12 \quad x_2 = 6$).

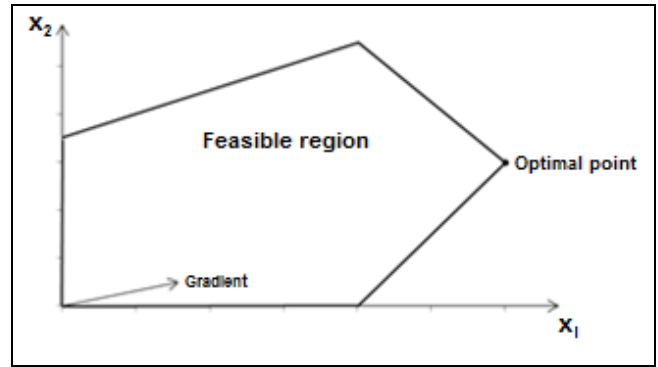


Figure 1. Polyhedron, gradient and optimal point

Initialisation

- Interior point (1) was used as P_0 so that $P_0 \in Ax < b$
- Gradient vector d was $\nabla = (3 \ 1)$ and $n-1$ orthogonal vectors to d had to be defined. $O_1 = (-1 \ 3)$ was chosen in this case.
- $n-1$ parametric straight lines were defined. The equation for the straight line was:

$$P_0 + O_1 t = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 3 \end{bmatrix} t, \text{ this was } P_0 + O_1 t: \begin{cases} x_1 = 1 - t \\ x_2 = 1 + 3t \end{cases}$$

- The $n-1$ parametric straight lines' intersection with hyperplanes π_j was found corresponding values of

$$t_{ij} = \frac{b_j - f_j \cdot P_0}{f_j \cdot O_i} \text{ were calculated:}$$

$$t_{11} = \frac{7 - \begin{bmatrix} -1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} -1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix}} = \frac{13}{7} \quad t_{12} = \frac{12 - \begin{bmatrix} 3 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} 3 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix}} = -\frac{23}{9}$$

$$t_{13} = \frac{21 - \begin{bmatrix} 5 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} 5 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix}} = \frac{75}{7} \quad t_{14} = \frac{0 - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix}} = 1$$

$$t_{15} = \frac{0 - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 3 \end{bmatrix}} = -\frac{1}{3}$$

- Values of t were chosen as follows:

$$t_i^{pos} = \min_{t_j > 0} (t_j) = \min\left(\frac{13}{7}, \frac{75}{7}, 1\right) = 1 \text{ and}$$

$$t_i^{neg} = \min_{t_j < 0} (|t_j|) = \min\left(\left|-\frac{23}{9}\right|, \left|-\frac{1}{3}\right|\right) = \frac{1}{3}$$

- The coordinates of the intersections with the hyperplanes of the polyhedron were:

$$\pi_4: \begin{cases} x_1 = 1 - 1 \cdot 1 = 0 \\ x_2 = 1 + 3 \cdot 1 = 4 \end{cases}$$

$$\pi_5 : \begin{cases} x_1 = 1 - 1 \cdot \frac{1}{3} = \frac{2}{3} \\ x_2 = 1 + 3 \cdot \frac{1}{3} = 2 \end{cases}$$

- The centroid was calculated as:

$$\mathbf{C}_0 = \begin{bmatrix} 0 + \frac{4}{3} & 4 + 0 \\ \frac{2}{2} & \frac{2}{2} \end{bmatrix} = \begin{bmatrix} \frac{4}{3} & 2 \end{bmatrix}; \text{ its location is shown in}$$

Figure 2.

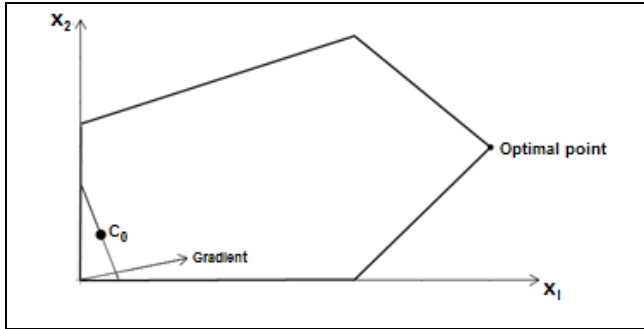


Figure 2. Centroid C0

- Straight line $\mathbf{C}_0 + \mathbf{d} t$: $\begin{cases} x_1 = \frac{2}{3} + 3t \\ x_2 = 2 + 1t \end{cases}$ was constructed

and the intersection with the five hyperplanes π_j was found by

calculating: $t_j = \frac{b_j - \mathbf{f}_j \cdot \mathbf{C}_0}{\mathbf{f}_j \cdot \mathbf{d}} \forall j$ led to:

$$t_1 = \frac{7 - \begin{bmatrix} -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} & 2 \end{bmatrix}}{\begin{bmatrix} -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \end{bmatrix}} = -\frac{32}{3} \quad t_2 = \frac{12 - \begin{bmatrix} 3 & -1 \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} & 2 \end{bmatrix}}{\begin{bmatrix} 3 & -1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \end{bmatrix}} = \frac{26}{7}$$

$$t_3 = \frac{21 - \begin{bmatrix} 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} & 2 \end{bmatrix}}{\begin{bmatrix} 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \end{bmatrix}} = \frac{218}{57} \quad t_4 = \frac{0 - \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} & 2 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \end{bmatrix}} = -2$$

$$t_5 = \frac{0 - \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} & 2 \end{bmatrix}}{\begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \end{bmatrix}} = -2$$

- $t_{\min} = \min_{t_j > 0} (t_j)$ was found. In this case

$$t_{\min} = \min_{t_j > 0} \left(\frac{26}{7} \quad \frac{218}{57} \right) = \frac{26}{7}$$

Coordinates \mathbf{F}_1 of the straight line with t_{\min} on the k -th of the polyhedron were calculated:

$$\mathbf{F}_1 = \mathbf{C}_0 + \mathbf{d} t_{\min} = \begin{pmatrix} \frac{2}{3} & 2 \end{pmatrix} + \begin{pmatrix} 3 & 1 \end{pmatrix} \frac{26}{7} = \begin{pmatrix} \frac{248}{21} & \frac{40}{7} \end{pmatrix}$$

As seen in Figure 3, point \mathbf{F}_1 was located on one of the faces of

polyhedron and it came close to the optimal point; the main cycle (iterative) then began.

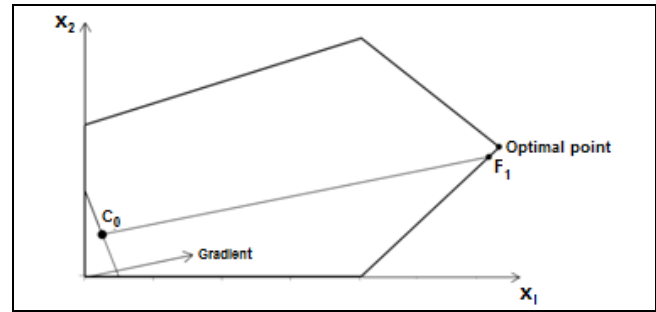


Figure 3. Gradient projection

Main cycle

A problem in R^2 is presented to illustrate each step in the algorithm and graphically visualise Point \mathbf{F}_1 was located on face

$\pi_2 : \frac{3}{2} x_1 - x_2 = 12$ resulting in $\mathbf{f}_2 = \begin{bmatrix} 3 & -1 \end{bmatrix}$. The first step was to find $n-1$ vectors $\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_{n-1}$ so that:

They were orthogonal to the gradient of objective function \mathbf{d}

The sign $(\mathbf{F}_1 \mathbf{C}_0 \cdot \mathbf{f}_2) = \text{sign} (\mathbf{O}_1 \cdot \mathbf{f}_2)$ or zero.

Selecting vector $\mathbf{O}_1 = \begin{bmatrix} -1 & 3 \end{bmatrix}$, led to:

$$\mathbf{F}_1 \mathbf{C}_0 = \begin{bmatrix} -\frac{234}{21} & -\frac{26}{7} \end{bmatrix} \Rightarrow$$

$$\mathbf{F}_1 \mathbf{C}_0 \cdot \mathbf{f}_2 = \begin{bmatrix} -\frac{234}{21} & -\frac{26}{7} \end{bmatrix} \cdot \begin{bmatrix} 3 & -1 \end{bmatrix} = -13$$

$\mathbf{O}_1 \cdot \mathbf{f}_2 = \begin{bmatrix} -1 & 3 \end{bmatrix} \cdot \begin{bmatrix} 3 & -1 \end{bmatrix} = -\frac{9}{2}$, thus the second condition was satisfied.

The intersection of each straight line $\mathbf{F}_i + \mathbf{O}_i t$ with each hyperplane was found (excluding the plane π_2).

- The t values indicated that the intersections of straight line

$$\mathbf{F}_1 + \mathbf{O}_1 t : \begin{cases} x_1 = \frac{248}{21} - t \\ x_2 = \frac{40}{7} + 3t \end{cases}$$

where:

$$t_{11} = \frac{7 - \begin{bmatrix} -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{248}{21} & \frac{40}{7} \end{bmatrix}}{\begin{bmatrix} -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 \end{bmatrix}} = \frac{302}{147}$$

$$t_{13} = \frac{21 - \begin{bmatrix} 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{248}{21} & \frac{40}{7} \end{bmatrix}}{\begin{bmatrix} 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 \end{bmatrix}} = \frac{44}{147}$$

$$t_{14} = \frac{0 - \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{248}{21} & \frac{40}{7} \end{bmatrix}}{\begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 \end{bmatrix}} = -\frac{248}{21}$$

$$t_{15} = \frac{0 - [0 \ 1] \cdot \begin{bmatrix} 248 & 40 \\ 21 & 7 \end{bmatrix}}{[0 \ 1] \cdot [1 \ -3]} = \frac{40}{21}$$

- The minimum positive value for t was chosen:

$$t_{\min} = \min_{t_j > 0} \left(\frac{302}{147} \quad \frac{44}{147} \quad \frac{40}{21} \right) = \frac{44}{147}$$

- Corresponding coordinates of intersection $Q_1 = F_1 + O_1 t_{\min}$ were calculated using t_{\min} , as seen in Figure 4:

$$Q_1 : \begin{cases} x_1 = \frac{248}{21} - \frac{44}{147} = \frac{35532}{3087} \\ x_2 = \frac{40}{7} + 3 \left(\frac{44}{147} \right) = \frac{6804}{1029} \end{cases}$$

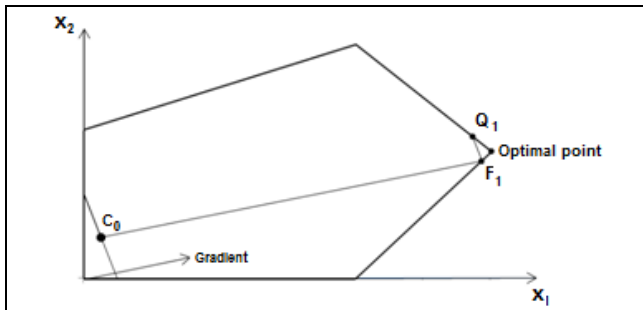


Figure 4. Orthogonal projection (Q_1)

The new centroid value was calculated: $C_1 = \frac{Q_1 + F_1}{2}$ (Figure 5).

$$C_1 = \begin{bmatrix} \frac{35532}{3087} + \frac{248}{21} & \frac{6804}{1029} + \frac{40}{7} \\ \frac{83986}{7203} & \frac{44394}{7203} \end{bmatrix} = \begin{bmatrix} \frac{83986}{7203} & \frac{44394}{7203} \end{bmatrix}$$

- The straight line was constructed

$$C_1 + d t : \begin{cases} x_1 = \frac{83986}{7203} + 3t \\ x_2 = \frac{44394}{7203} + t \end{cases}$$

and its intersection with the five hyperplanes was found

from $t_j = \frac{b_j - f_j \cdot C_1}{f_j \cdot d}$ by calculating:

$$t_1 = \frac{7 - \begin{bmatrix} -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 83986 & 44394 \\ 7203 & 7203 \end{bmatrix}}{\begin{bmatrix} -1 & 1 \end{bmatrix} \cdot [3 \ 1]} = -\frac{40}{3}$$

$$t_2 = \frac{12 - \begin{bmatrix} 3 & -1 \end{bmatrix} \cdot \begin{bmatrix} 83986 & 44394 \\ 7203 & 7203 \end{bmatrix}}{\begin{bmatrix} 3 & -1 \end{bmatrix} \cdot [3 \ 1]} = \frac{66}{343}$$

$$t_3 = \frac{21 - \begin{bmatrix} 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 83986 & 44394 \\ 7203 & 7203 \end{bmatrix}}{\begin{bmatrix} 5 & 1 \end{bmatrix} \cdot [3 \ 1]} = \frac{22}{399}$$

$$t_4 = \frac{0 - [1 \ 0] \cdot \begin{bmatrix} 83986 & 44394 \\ 7203 & 7203 \end{bmatrix}}{[1 \ 0] \cdot [3 \ 1]} = -\frac{1714}{441}$$

$$t_5 = \frac{0 - [0 \ 1] \cdot \begin{bmatrix} 83986 & 44394 \\ 7203 & 7203 \end{bmatrix}}{[0 \ 1] \cdot [3 \ 1]} = -\frac{302}{49}$$

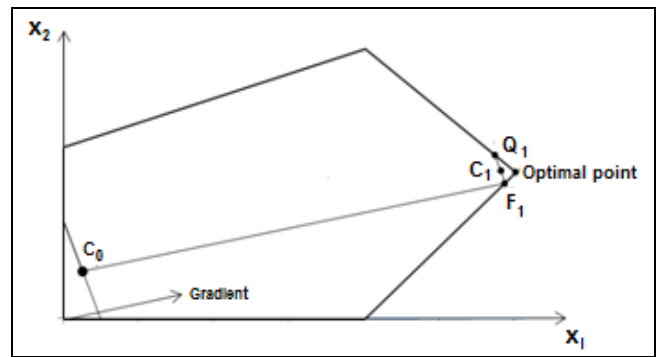


Figure 5. New centroid C_1

- Defining $t_{\min} = \min(t_2, t_3) = \min\left(\frac{66}{343} \quad \frac{22}{399}\right) = t_3 = \frac{22}{399}$

- Calculating the coordinates of point F_2 (shown in Figure 6) on the k -th face of the polyhedron:

$$F_2 = C_1 + d t_{\min} \in \pi_3$$

$$F_2 = C_1 + d t_{\min} = \begin{pmatrix} \frac{83986}{7203} & \frac{44394}{7203} \end{pmatrix} + (3 \ 1) \frac{22}{399} = (11.8253 \ 6.2184)$$

- Comparing the coordinates of F_{i-1} with F_i .

$$(11.8095 \ 5.7143) \text{ cf } (11.8253 \ 6.2184)$$

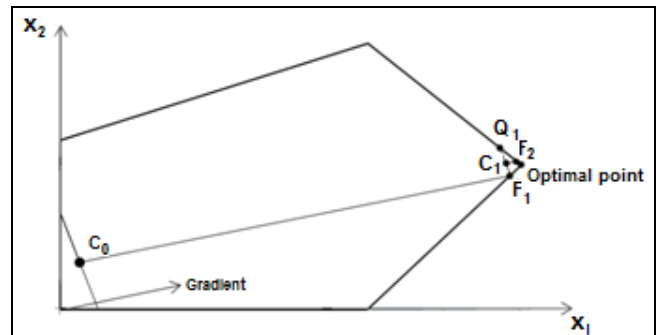


Figure 6. Projection of the gradient from C_1 to F_2

The Euclidean distance of points F_1 and F_2 to the optimal solution were 0.3063 and 0.2797, respectively. This iterative procedure was repeated until the established precision criterion became satisfied.

Conclusions and future research

A new algorithm for solving linear programming problems which does not require slack or excess variables, inverting matrices and the non-negativity of the variables has been presented. The algorithm's complexity and comparison with other methods are important topics for future research currently being investigated by the authors, as well as variations in the algorithm, such as transformation of variables and alternative ways of projection to increase convergence speed.

References

- Bazaraa, M. Jarvis, J. & Sherali, H. Programación lineal y flujo en redes, 2 ed en español, México, Limusa, 1998.
- Cho, G. An interior-point algorithm for linear optimization based on a new barrier function. Applied Mathematics and Computation, Vol 218, No 2, 2011, pp. 386-395.

- Cottle, R W. George B. Dantzig: a legendary life in mathematical programming. *Mathematical Programming*, Vol 105, No 1, 2006, pp. 1-8.
- Karmarkar, N. New Polynomial-Time algorithm for linear programming. *Combinatorica*, Vol 4, No 4, 1984, pp. 373-395.
- Khachiyan, L. On the exact solution of systems of linear inequalities and linear programming problems. *URSS Computational mathematics and Mathematical Physics*, Vol 22, No 4, 1982, pp. 239-242.
- Kim, M. Lee, Y. Cho, G. An adaptive-step primal-dual interior point algorithm for linear optimization. *Nonlinear Analysis*, Vol 71, 2009, pp. 2305-2315.
- Mizuno, S. A predictor-corrector infeasible-interior-point algorithm for linear programming. *Operations Research Letters*, Vol 16, No 2, 1994, pp. 61-66.
- Naseri, R. Valinejad, A. An extended variant of Karmarkar's interior point algorithm. *Applied Mathematics and Computation*, Vol 184, 2007, pp. 737-742.
- Norman D. Curet, N. A primal-dual simplex method for linear programs. *Operations Research Letters*, Vol 13, No 4, 1993, pp. 233-237.
- Pan, P. A projective Simplex algorithm using LU decomposition. *Computers and mathematics with applications*, Vol 39, 2000, pp. 187-208.
- Powell, M.J.D. On the number of iterations of Karmarkar's algorithm for linear programming. *Mathematical Programming*, Vol 62, 1993, pp. 153-197.
- Winston, W. *Investigación de Operaciones, aplicaciones y algoritmos*, 4 ed en español, México, Thompson, 2005, pp. 597-604.
- Zhang, L. Xu, Y. A full-Newton step interior-point algorithm based on modified Newton direction. *Operations Research Letters*, Vol 39, 2011, pp. 318-322.