LINEAR AND NONLINEAR PROGRAMMING: MATH 404

Revised Simplex Method

University of Science and Technology at Zewail City

Fall 2022

Prepared By
Mariam Wagdy
201801585

Supervised By
Dr. Ahmed Abdelamea

I. Table of Contents

II. Motivation

The simplex method is of fundamental importance for linear programming problems. It has numerous applications in the world of finances. Despite that, it includes a lot row operations “than needed” that consumes time and computational effort: since only the iterations in the entering and leaving variable are needed. Moreover, it is not practical for large problems since the simplex tableau needs to be calculated and stored on each iteration. In order to avoid the quick propagation of roundoff errors and digit loss, the revised simplex method has been introduced. In some large LP problems, the simplex method may lead to “serious loss of accuracy” (Taha, 2017).

Revised simplex algorithm takes the advantage of sparce matrices unlike the conventional simplex, speeding up the computation much and increasing its efficiency. (Taha, 2017 and Morgan, 1997). The iterative steps are the same as the simplex method. But instead of computing each row, the matrices are manipulated as a whole. Hence, the entire simplex tableau can be constructed by the knowledge of the basis at first and the computing the inverse of basis matrix (Taha, 2017).

III. Theory

The theory is obtained from Taha (2018).

Linear programming problem in standard form

$$\text{Min} \quad z = \mathbf{C^T X}$$

$$\text{S.t.} \quad \mathbf{AX} = \mathbf{b}$$
$$\mathbf{X} \geq \mathbf{0}$$
$$\mathrm{C}, \mathrm{X} \in \mathbb{R}^{\mathbf{n}}$$
$$\mathrm{b} \in \mathbb{R}^{\mathbf{m}}$$

Then,

$$\sum_{j=1}^{n} \mathbf{P}_j x_j = \mathbf{b}$$

Where $\boldsymbol{P}_j$ constitutes the *j*th column of $\boldsymbol{A}$. If these columns are linearly independent, they spam $\mathbb{R}^{\boldsymbol{n}}$, and hence there is a unique solution. Define $\boldsymbol{B}$ as the subset of *m* vectors in $\boldsymbol{A}$. Consequently, matrix $\boldsymbol{B}$ is nonsingular.

$$\mathbf{B}\mathbf{X}_\mathrm{B} = \mathbf{b}$$

Then

$$\mathbf{X}_\mathrm{B} = \mathbf{B}^{-1}\mathbf{b}$$

If $\mathbf{B}^{-1}\mathbf{b} \geq 0$, then $\mathbf{X}_\mathrm{B}$ is feasible. And the remaining *n-m* variables are nonbasic at the initial point.

Generalized Simplex Tableau in Matrix Form

$$\begin{pmatrix} 1 & -\mathbf{C} \\ \mathbf{0} & \mathbf{A} \end{pmatrix} \begin{pmatrix} z \\ \mathbf{X} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{b} \end{pmatrix}$$

Suppose that $\mathbf{B}$ is a feasbible basis for $\mathbf{AX} = \mathbf{b}$, and $\mathbf{X_B}$ as the corresbpnding vector of basic variables and $\boldsymbol{C_B}$ as its objective vector, the the solution can be

$$\begin{pmatrix} z \\ \mathbf{X_B} \end{pmatrix} = \begin{pmatrix} 1 & -\mathbf{C_B} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \mathbf{b} \end{pmatrix}$$

By using inversion of partationed matrices,

$$\begin{pmatrix} z \\ \mathbf{X_B} \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{C_B B^{-1}} \\ \mathbf{0} & \mathbf{B^{-1}} \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{b} \end{pmatrix} = \begin{pmatrix} \mathbf{C_B B^{-1} b} \\ \mathbf{B^{-1} b} \end{pmatrix}$$

Then the complete simplex tabluje an be computed from

$$\begin{pmatrix}1 & \mathbf{C_B B^{-1}}\\ \mathbf{0} & \mathbf{B^{-1}}\end{pmatrix}\begin{pmatrix}1 & \mathbf{-C}\\ \mathbf{0} & \mathbf{A}\end{pmatrix}\begin{pmatrix}z\\ \mathbf{X}\end{pmatrix}=\begin{pmatrix}1 & \mathbf{C_B B^{-1}}\\ \mathbf{0} & \mathbf{B^{-1}}\end{pmatrix}\begin{pmatrix}0\\ \mathbf{b}\end{pmatrix}=\begin{pmatrix}\mathbf{C_B B^{-1}b}\\ \mathbf{B^{-1}b}\end{pmatrix}$$

Further reducing to

$$\begin{pmatrix}1 & \mathbf{C_B B^{-1}A - C}\\ \mathbf{0} & \mathbf{B^{-1}A}\end{pmatrix}\begin{pmatrix}z\\ \mathbf{X}\end{pmatrix}=\begin{pmatrix}\mathbf{C_B B^{-1}b}\\ \mathbf{B^{-1}b}\end{pmatrix}$$

Given the jth vector of Pj of A, the simplex coulmn for that vector is

Basic	$\mathbf{x_j}$	**Solution**
z	$\mathbf{C_B B^{-1}P_j} - c_j$	$\mathbf{C_B B^{-1}b}$
$\mathbf{X_B}$	$\mathbf{B^{-1}P_j}$	$\mathbf{B^{-1}b}$

By inspection, we can see that only the matrix $\mathbf{B^{-1}}$ changes in the simplex tablue. That means the entire talue can be constructed from the orignal problem if $\mathbf{B^{-1}}$ is calculated. But to optain **B**, we need to define the basic columns of **X**, which are eventually $\mathbf{X_B}$.

On the contrary to the simplex megthod, this way we can avoid roundoff error and the excessive memory and resources in computing by computing $\mathbf{B^{-1}}$ from the original constraint columns.

Hence, from the avobe tablue, any simplex iteration can be represented by the following equations:

$$z + \sum_{j=1}^{n}(z_j - c_j)x_j = \mathbf{C_B B^{-1}b}$$

$$(\mathbf{X_B})_i + \sum_{j=1}^{n}\left(\mathbf{B^{-1}P_j}\right)_i x_j = (\mathbf{B^1 b})_i$$

Where

$$(z_j - c_j) = \mathbf{C_B B^{-1}P_j} - c_j$$

Where *i* represents the ekement index in the vector *j*.

Optimality condition

For minimization, the condition is $(z_j - c_j) > 0$. Thus, the entering vector is selected as the nonbasic vector with the most Positive $(z_j - c_j)$ in case of minimization. The opposite is true for maximzation.

Feasibility Condition

Given the entering vector p_j as determined by the optimality condition, the constraint equations reduce to

$$(\mathbf{X_B})_i = (\mathbf{B^1 b})_i - \left(\mathbf{B^{-1}P_j}\right)_i x_j$$

The purpose is to increase x_j above zero because the other n-1 varables are zero. The limit to that increase is the following condition:

$$(\mathbf{X_B})_i = (\mathbf{B^1 b})_i - \left(\mathbf{B^{-1}P_j}\right)_i x_j \geq \mathbf{0}$$

If $\left(\mathbf{B^{-1}P_j}\right)_i > \mathbf{0}$ for at least one *i*, then

$$x_j = \min\left\{\frac{(\mathbf{B^{-1}b})_i}{\left(\mathbf{B^{-1}P_j}\right)_i} \mid \left(\mathbf{B^{-1}P_j}\right)_i > \mathbf{0}\right\}$$

Suppose that $(\boldsymbol{X_B})_k$ is the basic variable that corresponds to the minimum ratio. Then, $\boldsymbol{p}_k$ is the leaving vector, and its associated (basic) variable must become nonbasic (zero) in the next iteration.

IV. Algorithm

Step 1. Put the problem in standard form and choose a feasible starting point. Choose the basic variables and let **B** and $\boldsymbol{C_B}$ [1]be its associated basis and objective coefficients vector, respectively.

Step 2. Compute the inverse $\boldsymbol{B^{-1}}$ of the basis **B**

Step 3. For each nonbasic vector $\mathbf{p_j}$, compute

$$(z_j - c_j) = \mathbf{C_B B^{-1} P_j} - c_j$$

If $z_j - c_j \geq 0$ in maximization (≤ 0 in minimization) for all nonbasic vectors, stop; the optimal solution is $\mathbf{X_B = B^{-1} b}, z = \mathbf{C_B X_B}$.

Else, determine the entering vector $\mathbf{p_j}$ having the most negative (positive) $z_j - c_j$ in case of maximization (minimization) among all nonbasic vectors.

Step 4. Compute $\mathbf{B^{-1} P_j}$. If all the elements of $\mathbf{B^{-1} P_j}$ are negative or zero, stop; the solution is unbounded. Else, use the ratio test to determine the leaving vector $\mathbf{p_i}$.

Step 5. Form the next basis by replacing the leaving vector $\mathbf{p_i}$ with the entering vector $\mathbf{p_j}$ in the current basis **B**. Go to step 1 to start a new iteration. (Taha, 2017)

[1] If the problem involves artificial variables, replace $\mathbf{C_B}$ by $\mathbf{D_B}$ where **D** is the vector including the coefficients for the problem $w = \mathbf{D^T X}$ and let w include only the artificial variables.

V. Implementation

Example 1:

$$Max \quad z = 2x_1 + x_2$$

$$S.t. \quad 3x_1 + 4x_2 \leq 6$$
$$6x_1 + x_2 \leq 3$$
$$x_1, x_2 \geq 0$$

Put the problem in standard form

$$Min \quad z = -2x_1 - x_2$$

$$S.t. \quad 3x_1 + 4x_2 + x_3 = 6$$
$$6x_1 + x_2 + x_4 = 3$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Hence $n = 4, m = 2$

$$\boldsymbol{X} = [\, x_1\ x_2\ x_3\ x_4]$$
$$\boldsymbol{C} = [-2\ -1\ 0\ 0]$$
$$\boldsymbol{A} = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 6 & 1 & 0 & 1 \end{bmatrix}$$
$$\boldsymbol{b} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

Iteration 0:

$$\boldsymbol{X_{B0}} = [\, x_3\ x_4]$$
$$\boldsymbol{C_{B0}} = [\, 0\ 0\ ]$$
$$\boldsymbol{B_0} = (\boldsymbol{P_3}, \boldsymbol{P_4}) = \boldsymbol{I}$$
$$\boldsymbol{B_0^{-1}} = \boldsymbol{I}$$

Thus

$$\boldsymbol{X_{B0}} = \boldsymbol{B_0^{-1}b} = [\, \mathbf{6}\ \mathbf{3}\ ]$$
$$z = \boldsymbol{C_{B0}X_{B0}} = \mathbf{0}$$

Optimality Condition:

$$\boldsymbol{C_{B0}B_0^{-1}} = [\mathbf{0}\ \mathbf{0}]$$
$$\{z_j - c_j\}_{j=1,2} = \boldsymbol{C_{B0}B_0^{-1}}[\, \boldsymbol{P_1P_2}] - [c_1\ c_2] = [2\ 1]$$

Looking for most positive vector, $\mathbf{P_1}$ is the entering vector.

Feasibility Condition:

$\boldsymbol{X_{B0}} = [\boldsymbol{x_3}\ \ \boldsymbol{x_4}]^{\boldsymbol{T}}$

$\boldsymbol{B_0^{-1}P_1} = [\mathbf{3}\ \mathbf{6}]^{\boldsymbol{T}}$

Hence,

$$x_1 = min\left\{\frac{6}{3}, \frac{3}{6}\right\} = 1/2$$

Then P_4 becomes the leaving vector

Basic	x_1	x_2	x_3	x_4	*Solution*
x_3	*3*				*6*
x_4	*6*				*3*
$-z$	*-2*	*-1*	*0*	*0*	*0*

Iteration 1:

$$X_{B1} = [\, x_3 \; x_1]$$
$$C_{B1} = [\, 0 \; -2\,]$$
$$B_1 = (P_3, P_1) = \begin{bmatrix} 1 & 3 \\ 0 & 6 \end{bmatrix}$$
$$B_1^{-1} = \begin{bmatrix} 1 & -1/2 \\ 0 & 1/6 \end{bmatrix}$$

Thus

$$X_{B1} = B_1^{-1} b = \begin{bmatrix} 1 & -1/2 \\ 0 & 1/6 \end{bmatrix} \begin{bmatrix} 6 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{9}{2} \\ \frac{1}{2} \end{bmatrix}$$

$$-z = C_{B1} X_{B1} = -1$$

Optimality Condition:

$$C_{B1} B_1^{-1} = [0 \; -1/3]$$
$$\{z_j - c_j\}_{j=4,2} = C_{B1} B_1^{-1} [\, P_4 P_2] - [c_4 c_2] = \left[-\frac{1}{3} \; \frac{2}{3}\right]$$

Thus $\mathbf{P_2}$ is the entering vector.

Feasibility Condition:

$$X_{B1} = [x_3 \;\; x_1]^T$$

$$B_1^{-1} P_2 = \left[\frac{7}{2} \;\; \frac{1}{6}\right]^T$$

Hence,

$$x_2 = min\left\{\frac{9/2}{7/2}, \frac{1/2}{1/6}\right\} = \frac{9}{7}$$

Then P_3 becomes the leaving vector.

Iteration 2:

$$X_{B2} = [\, x_2 \; x_1]$$

$$\boldsymbol{C_{B2}} = [\,-1 \;\; -2\,]$$
$$\boldsymbol{B_2} = (\boldsymbol{P_2}, \boldsymbol{P_1}) = \begin{bmatrix} \boldsymbol{4} & \boldsymbol{3} \\ \boldsymbol{1} & \boldsymbol{6} \end{bmatrix}$$
$$\boldsymbol{B_2^{-1}} = \begin{bmatrix} \boldsymbol{2/7} & \boldsymbol{-1/7} \\ \boldsymbol{-1/21} & \boldsymbol{4/21} \end{bmatrix}$$

Thus

$$\boldsymbol{X_{B2}} = \boldsymbol{B_2^{-1}b} = \begin{bmatrix} \boldsymbol{2/7} & \boldsymbol{-1/7} \\ \boldsymbol{-1/21} & \boldsymbol{4/21} \end{bmatrix} \begin{bmatrix} \boldsymbol{6} \\ \boldsymbol{3} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\frac{9}{7}} \\ \boldsymbol{\frac{2}{7}} \end{bmatrix}$$

$$-z = \boldsymbol{C_{B2}X_{B2}} = -\boldsymbol{\frac{13}{7}}$$

Optimality Condition:

$$\boldsymbol{C_{B2}B_2^{-1}} = \left[-\boldsymbol{\frac{4}{21}} \;\; -\boldsymbol{\frac{5}{21}}\right]$$
$$\{z_j - c_j\}_{j=4,3} = \boldsymbol{C_{B2}B_2^{-1}}[\,\boldsymbol{P_4P_3}] - [c_4 c_3] = \left[-\frac{5}{21} \;\; -\frac{4}{21}\right]$$

Stop! Optimal solution is $\mathbf{X_B} = \begin{bmatrix} \frac{9}{7} \\ \frac{2}{7} \end{bmatrix}$ and $\boldsymbol{z} = \frac{13}{7}$

Graphical Method

Validation of previous result is obtained by using Desmos graphing calculator and setting $\boldsymbol{x_1}$ as $\boldsymbol{x}$ and $\boldsymbol{x_2}$ as $\boldsymbol{y}$. Where $\boldsymbol{a}$ is a slider to represent function contours.

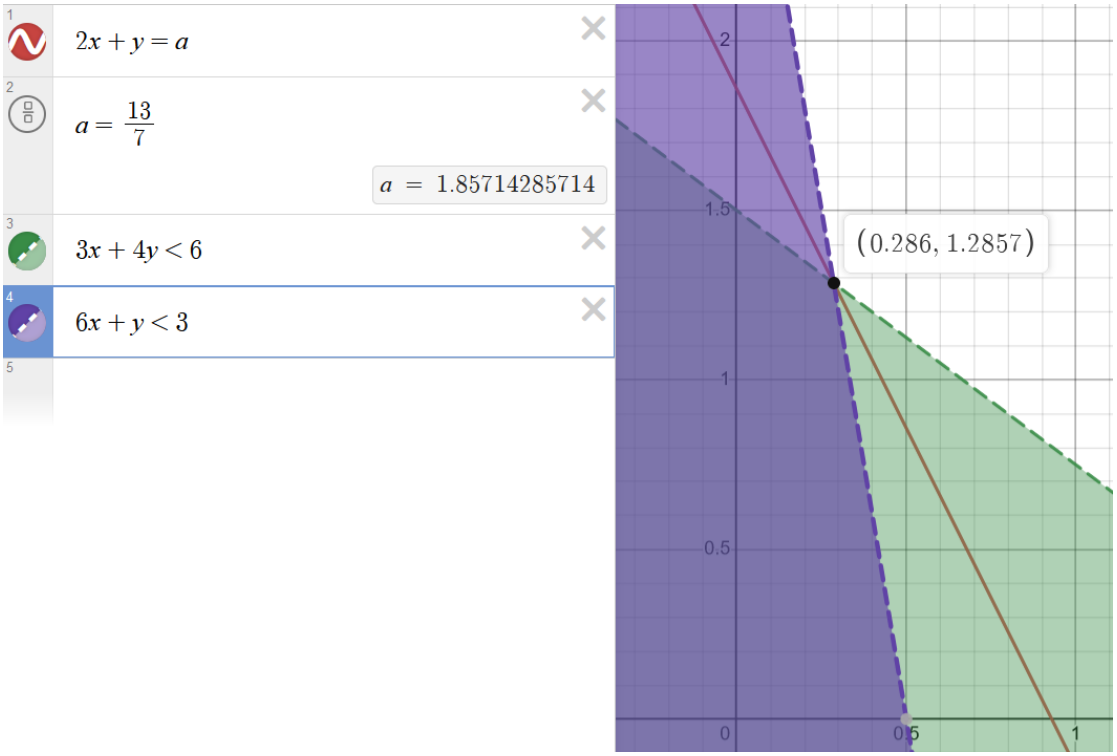


Figure 1 A graphical representation of the feasible region, highlighting the vertices and the optimum point with respect to the maximum objective function contour. (Desmos | Graphing Calculator, n.d.)

Example 2:

Part of the solution is obtained through my comprehension of Wolkowicz (2005).

$$Max \quad z = x_1 - 8x_2$$

$$\begin{aligned} s.t. \quad & 3x_1 + 2x_2 \geq 6 \\ & x_1 - x_2 \leq 6 \\ & 9x_1 + 7x_2 \leq 108 \\ & 3x_1 + 7x_2 \leq 70 \\ & 2x_1 - 5\,x_2 \geq -35 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Put the problem in standard form

$$Min \quad z = -x_1 + 8x_2$$

$$
\begin{aligned}
s.t.\quad & 3x_1 + 2x_2 - x_3 + y_1 = 6\\
& x_1 - x_2 + x_4 = 6\\
& 9x_1 + 7x_2 + x_5 = 108\\
& 3x_1 + 7x_2 + x_6 = 70\\
& -2x_1 + 5x_2 + x_7 = 35
\end{aligned}
$$

$$x_1, x_2, x_3, y_1, x_4, x_5, x_6, x_7 \geq 0$$

Minimize $\mathbf{w} = \mathbf{y_1} = \mathbf{D^T X}$

$$\boldsymbol{X} = [\,\boldsymbol{x_1}\ \boldsymbol{x_2}\ \boldsymbol{x_3}\ \boldsymbol{y}\ \boldsymbol{x_4}\ \boldsymbol{x_5}\ \boldsymbol{x_6}\ \boldsymbol{x_7}]^{\boldsymbol{T}}$$
$$\mathbf{D^T} = [\,\mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\,]$$

Then $n = 8, m = 5$ and we have 5 basic variables

$$C = [\mathbf{-1}\quad \mathbf{8}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}]$$

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{3} & \mathbf{2} & \mathbf{-1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{9} & \mathbf{7} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{3} & \mathbf{7} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{-2} & \mathbf{5} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} \mathbf{6} \\ \mathbf{6} \\ \mathbf{108} \\ \mathbf{70} \\ \mathbf{35} \end{bmatrix}$$

Iteration 0:

$$\boldsymbol{X_{B0}} = [\boldsymbol{y_1}\ \boldsymbol{x_4}\ \boldsymbol{x_5}\ \boldsymbol{x_6}\ \boldsymbol{x_7}]$$
$$\boldsymbol{D_{B0}} = [\ \mathbf{1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\,]$$
$$\boldsymbol{C_{B0}} = [\ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\,]$$
$$\boldsymbol{B_0} = (\boldsymbol{P_4}, \boldsymbol{P_5}, \boldsymbol{P_6}, \boldsymbol{P_7}, \boldsymbol{P_8}) = \boldsymbol{I}$$
$$\boldsymbol{B_0^{-1}} = \boldsymbol{I}$$

Thus

$$\boldsymbol{X_{B0}} = \boldsymbol{B_0^{-1}b} = [\,\mathbf{6}\ \ \mathbf{6}\ \ \mathbf{108}\ \ \mathbf{70}\ \ \mathbf{35}]$$
$$w = \boldsymbol{D_{B0}X_{B0}} = \mathbf{6}$$

Optimality Condition:

$$\boldsymbol{D_{B0}B_0^{-1}} = [\mathbf{1\ 0\ 0\ 0\ 0}]$$
$$\{z_j - d_j\}_{j=1,2,3} = \boldsymbol{D_{B0}B_0^{-1}}[\,\boldsymbol{P_1 P_2\ P_3}] - [d_1\ d_2\ d_3] = [3\quad 2\quad -1]$$

Looking for most positive vector, $\mathbf{P_1}$ is the entering vector.

Feasibility Condition:

$$\boldsymbol{X_{B0}} = [\boldsymbol{y_1}\ \boldsymbol{x_4}\ \boldsymbol{x_5}\ \boldsymbol{x_6}\ \boldsymbol{x_7}]^{\boldsymbol{T}}$$
$$\boldsymbol{B_0^{-1}P_1} = [\mathbf{3}\ \ \mathbf{1}\ \ \mathbf{9}\ \ \mathbf{3}\ \ \mathbf{-2}]^{\boldsymbol{T}}$$

Hence,

$$\boldsymbol{x_1 = min\left\{\frac{6}{3}, \frac{6}{1}, \frac{108}{9}, \frac{70}{3}, \frac{35}{-2}\right\} = 2}$$

Then $\boldsymbol{P_4}$ becomes the leaving vector ($\boldsymbol{y_1}$).

Iteration 1:

$$\boldsymbol{X_{B1}} = [\boldsymbol{x_1}\ \boldsymbol{x_4}\ \boldsymbol{x_5}\ \boldsymbol{x_6}\ \boldsymbol{x_7}]$$
$$\boldsymbol{D_{B1}} = [\ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ ]$$
$$\boldsymbol{C_{B1}} = [\ \mathbf{-1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ ]$$
$$\boldsymbol{B_1} = (\boldsymbol{P_1}, \boldsymbol{P_5}, \boldsymbol{P_6}, \boldsymbol{P_7}, \boldsymbol{P_8}) = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 9 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\boldsymbol{B_1^{-1}} = \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 & 0 \\ -3 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ \frac{2}{3} & 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus

$$\boldsymbol{X_{B1}} = \boldsymbol{B_1^{-1}b} = \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 & 0 \\ -3 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ \frac{2}{3} & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 6 \\ 108 \\ 70 \\ 35 \end{bmatrix} = [\ \mathbf{2}\ \ \mathbf{4}\ \ \mathbf{90}\ \ \mathbf{64}\ \ \mathbf{39}]^{\boldsymbol{T}}$$
$$w = \boldsymbol{D_{B1}X_{B1}} = \mathbf{0}$$

Stop.

Iteration 2:

$$C = [\mathbf{-1}\quad \mathbf{8}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}\quad \mathbf{0}]$$

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{3} & \mathbf{2-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{9} & \mathbf{7} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{3} & \mathbf{7} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{-2} & \mathbf{5} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} \mathbf{6} \\ \mathbf{6} \\ \mathbf{108} \\ \mathbf{70} \\ \mathbf{35} \end{bmatrix}$$

$$\boldsymbol{X_{B2}} = [\boldsymbol{x_1}\ \boldsymbol{x_4}\ \boldsymbol{x_5}\ \boldsymbol{x_6}\ \boldsymbol{x_7}]$$
$$\boldsymbol{C_{B2}} = [\ \mathbf{-1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ ]$$
$$\boldsymbol{B_2} = (\boldsymbol{P_1}, \boldsymbol{P_4}, \boldsymbol{P_5}, \boldsymbol{P_6}, \boldsymbol{P_7}) = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 9 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{B_2^{-1}} = \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 & 0 \\ -3 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ \frac{2}{3} & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{X_{B2}} = [\,\mathbf{2}\ \ \mathbf{4}\ \ \mathbf{90}\ \ \mathbf{64}\ \ \mathbf{39}]^{\boldsymbol{T}}$$

Optimality Condition:

$$\boldsymbol{C_{B2}B_2^{-1}} = \left[-\frac{\mathbf{1}}{\mathbf{3}}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\right]$$

$$\{z_j - c_j\}_{j=2,3} = \boldsymbol{C_{B2}B_2^{-1}}[\,\boldsymbol{P_2}\ \boldsymbol{P_3}] - [\,c_2\ c_3] = \left[\ \frac{-2}{3}\ \ \frac{1}{3}\right] - [8\ 0] = \left[-\frac{26}{3}\ \frac{1}{3}\right]$$

Looking for most positive vector, $\mathbf{P_3}$ is the entering vector.

Feasibility Condition:

$$\boldsymbol{X_{B2}} = [\boldsymbol{x_1}\ \boldsymbol{x_4}\ \boldsymbol{x_5}\ \boldsymbol{x_6}\ \boldsymbol{x_7}]^{\boldsymbol{T}}$$

$$\boldsymbol{B_2^{-1}P_3} = \left[\frac{-\mathbf{1}}{\mathbf{3}}\ \ \frac{\mathbf{1}}{\mathbf{3}}\ \ \mathbf{3}\ \ \mathbf{1}\ \ -\frac{\mathbf{2}}{\mathbf{3}}\right]^{\boldsymbol{T}}$$

Hence,

$$\boldsymbol{x_3} = \boldsymbol{min}\left\{-\ \ \frac{\mathbf{4}}{\frac{\mathbf{1}}{\mathbf{3}}}\ \ \frac{\mathbf{90}}{\mathbf{3}}\ \ \ \mathbf{64}\ \ -\right\} = \mathbf{12}$$

Then $\boldsymbol{x_4}$ becomes the leaving vector.

Iteration 3:

$$\boldsymbol{X_{B3}} = [\boldsymbol{x_1}\ \boldsymbol{x_3}\ \boldsymbol{x_5}\ \boldsymbol{x_6}\ \boldsymbol{x_7}]$$

$$\boldsymbol{C_{B3}} = [\,\mathbf{-1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\,]$$

$$\boldsymbol{B_3} = (\boldsymbol{P_1}, \boldsymbol{P_3}, \boldsymbol{P_5}, \boldsymbol{P_6}, \boldsymbol{P_7}) = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 9 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{B_3^{-1}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & -3 & 0 & 0 & 0 \\ 0 & -9 & 1 & 0 & 0 \\ 0 & -3 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{X_{B3}} = [\,\mathbf{6}\ \ \mathbf{12}\ \ \mathbf{54}\ \ \mathbf{52}\ \ \mathbf{47}]^{\boldsymbol{T}}$$

$$\boldsymbol{-z = -6}$$

Optimality Condition:

$$\boldsymbol{C_{B3}B_3^{-1}} = [\mathbf{0}\ \ \mathbf{-1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}]$$
$$\{z_j - c_j\}_{j=2,4} = \boldsymbol{C_{B3}B_3^{-1}}[\,\boldsymbol{P_2}\ \boldsymbol{P_4}] - [\,c_2\ c_4] = [1\ \ -1] - [8\ \ 0] = [-7\ -1]$$

Stop $\mathbf{X_B} = [\mathbf{6}\ \ \mathbf{12}\ \ \mathbf{54}\ \ \mathbf{52}\ \ \mathbf{47}]$, $\mathrm{z} = \mathbf{6}$

$$x_1 = 6, x_2 = 0\ , x_3 = 12, x_4 = 54, x_5 = 52, x_6 = 47$$

Graphical Method

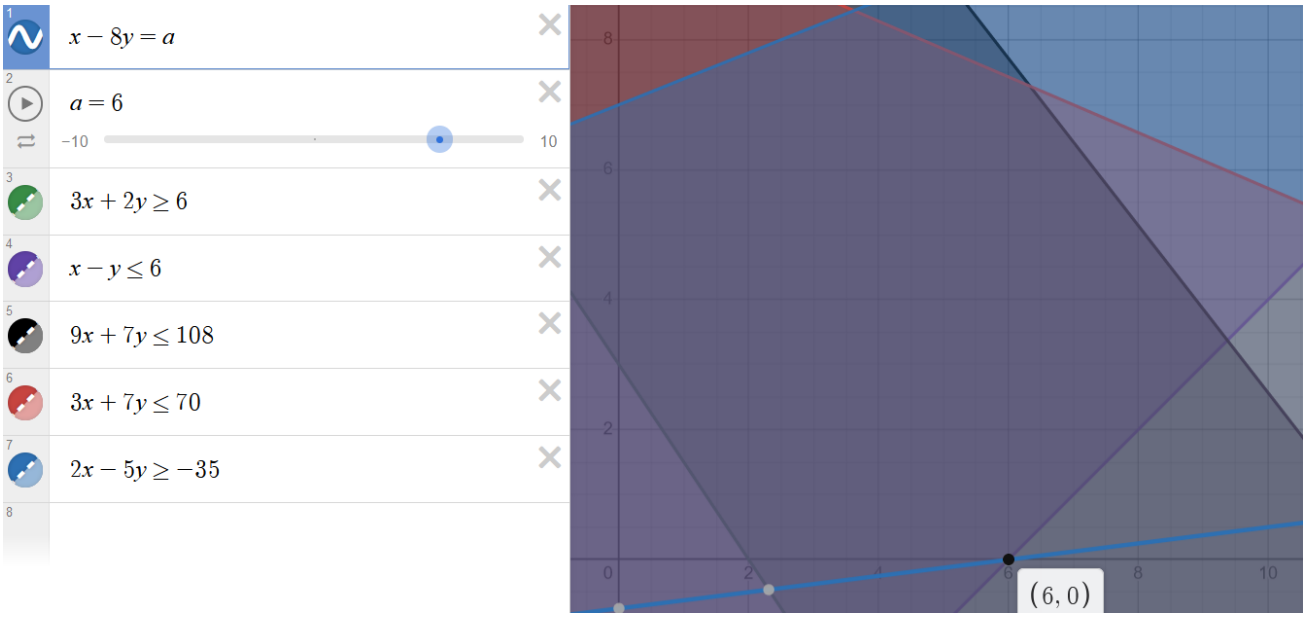


Figure 2 A graphical representation of the feasible region, highlighting the vertices and the optimum point with respect to the maximum objective function contour (Desmos | Graphing Calculator, n.d.)

VI. Applications

Application of the Revised Simplex Method to the Farm Planning Model

With regard to dividing up a farm's limited resources, such as land, labour, and capital, amongst alternative crop and livestock enterprises, the Farm Planning Model is used. It supports farmers in making decisions about what and how much will be produced in order to get the best returns from the resources they have or access to.

To tackle the optimization problem of allocating the limited resources to the products in a way that maximises revenues or, alternatively, minimises costs, the Farm Planning Model employs the linear programming method. (HUA, 1998)

Revised simplex method and its application for solving fuzzy linear programming problems

Artificial intelligence, computer science, control engineering, decision theory, expert systems, logic, management science, operations research, robotics, and other fields use fuzzy technologies.

The extended crisp LP problem can be efficiently solved using the revised simplex method in revision. (Nasseri et al., 2012)

Application Of Revised Simplex Method for Profit Maximization in A Paint Company

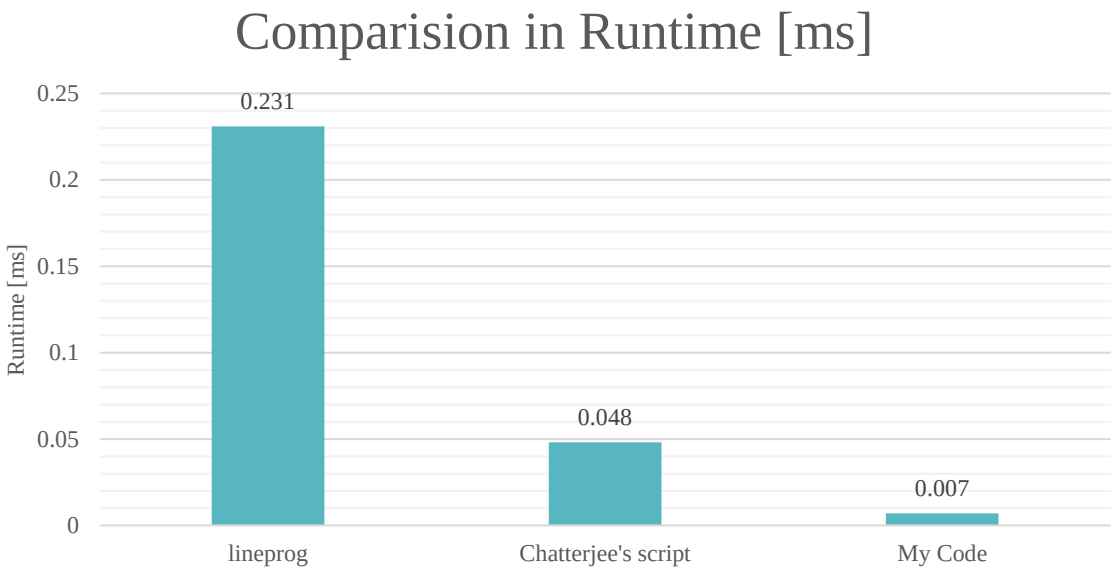
The paper utilizes the revised simplex method in optimizing profit. The problem provided information on fifteen (15) distinct paint types and their revenue. (Ikpang et al., 2021)

VII. Comparison with MATLAB built-in function

The comparison was carried between MATLAB's function *linprog* and a MATLAB script for the revised simples done by Chatterjee (2022) on the two examples presented earlier.

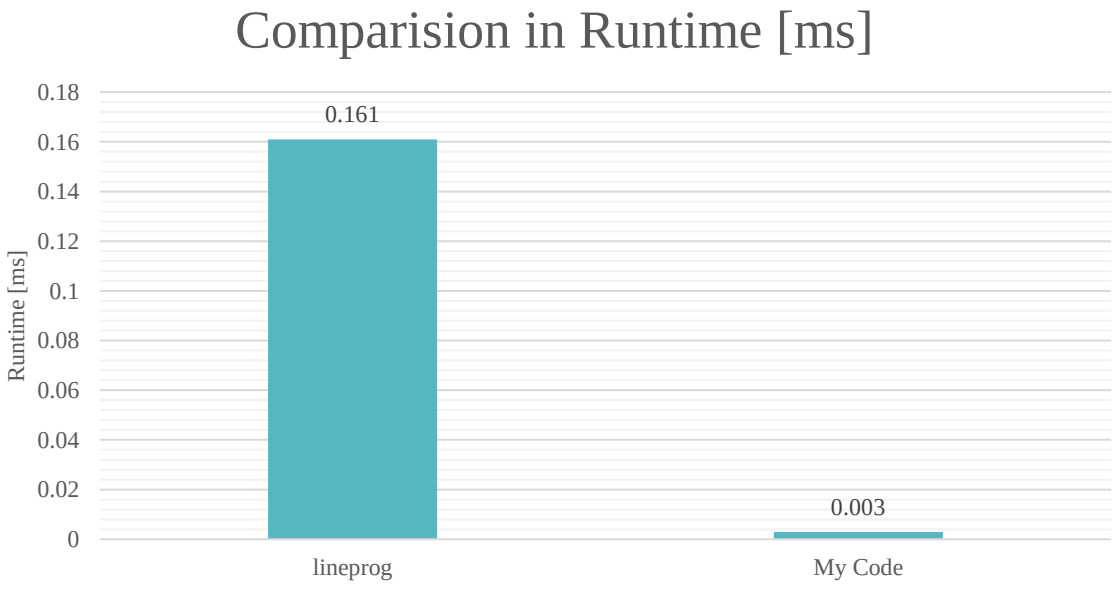
Example 1:

All gave the same results to the same precision. The only difference is the speed. The speed changed from time to time but all values were taken at the same time.



Example 2:

Linprog and my code gave the same result, while Chatterjee's script gave a wrong answer.



MATLAB Script

```
clear
clc
A=[-3 -2;1 -1;9 7 ;3 7;-2 5];
C=[-1 8];
b= [-6 ;6; 108; 70; 35];
s=[-1;-1;-1;-1;-1];
p=1;
key=1;
D=zeros(1,2*length(A));
for x=1:length(s)
    switch s(x)
        case -1
            point=size(A);
            A(x,point(2)+1)=1;
            C(length(C)+1)=0;
        case 0
            point=size(A);
            A(x, point(2)+1)=1;
            D(point(2)+1)=1;
            p=p+1;
            C(length(C)+1)=0;
            key=2;
        case 1
            point=size(A);
            A(x, point(2)+1)=-1;
            A(x, point(2)+2)=1;
            D(point(2)+2)=1;
            p=p+1;
            C(length(C)+2)=0;
            key=2;
        otherwise
            disp(" wrong input");
    end
end
[m,n]=size(A); %% m: number of constraints (basic variables)
               %%n: number of vartiables
k=n-m;         %% number of nonbasic variables
o=1;
o2=1;
%%%%% getting basis%%%%%
for x=1:n
    if (nnz(A(:,x)) == 1) && (max( A(:,x)) == 1)
        B(:,o)=A(:,x);
        CB(o)=C(x);
        DB(o)=D(x);
        L(o)=x; %% vector of basic elements indecies
        o=o+1;
```

```
48 -        else
49 -            J(o2)=x; %% vector of nonbasic elements indecies
50 -            o2=o2+1;
51 -        end
52 -    end
53
54 -    while (key)
55 -        switch key
56 -            case 1
57 -                B_inv=inv(B);
58 -                XB=B_inv*b;
59 -                z=-CB*XB;
60
61 -                for j=1:n-m
62 -                    zj_cj(j)=CB*B_inv*A(:,J(j))-C(J(j));
63 -                end
64 -                if all(zj_cj<=0)
65 -                    disp("solution is reached");
66 -                    disp("indeces of basic elements are");
67 -                    disp(L);
68 -                    disp("their value is");
69 -                    disp(XB);
70 -                    disp("optimum value is:");
71 -                    disp(z);
72 -                    break;
73 -                end
74 -                [entering,I]= max(zj_cj);%% get index of maximum positive ratio
75 -                B_invPj=B_inv*A(:,I);
76
77 -                if any(B_invPj > 0)
78 -                    M=(B_inv*b)./B_invPj;
79 -                    [leaving, E]=min(M);
80 -                    temp=J(I);
81 -                    J(I)=L(E); %% switching indeces between leaving and entering
82 -                    L(E)=temp; %%vectors in basic and nonbasic indeces vectors
83 -                    for x=1:length(L)
84 -                        B(:,x)=A(:,L(x));
85 -                        CB(x)=C(L(x));
86 -                    end
87 -                else
```

```
                disp(" solution is unbounded");
                break;
            end

        case 2
            B_inv=inv(B);
            XB=B_inv*b;
            w=-DB*XB;

            for j=1:n-m
                wj_dj(j)=DB*B_inv*A(:,J(j))-D(J(j));
            end
            if all(wj_dj<=0)
                disp("solution is reached");
                disp("indeces of basic elements are");
                disp(L);
                disp("their value is");
                disp(XB);
                disp("optimum value is:");
                disp(w);
            end
            [entering,I]= max(wj_dj);%% get index of maximum positive ratio
            B_invPj=B_inv*A(:,I);

            if any(B_invPj > 0)
                M=(B_inv*b)./B_invPj;
                [leaving, E]=min(M);
                temp=J(I);
                J(I)=L(E); %% switching indeces between leaving and entering
                L(E)=temp; %%vectors in basic and nonbasic indeces vectors
                for x=1:length(L)
                    B(:,x)=A(:,L(x));
                    DB(x)=D(L(x));
                    CB(x)=C(L(x));
                end
            else
                disp(" solution is unbounded");
                break;
            end
            if w==0
                key=1;
            end
    end
end
tic;toc
```

VIII. Conclusion

In conclusion, the revised simplex is of great importance and use in the world of cost and profit. It is more practical and efficient in large problems than the traditional method. However, it is advised to avoid using the inverse of a matrix, because it would be very costly computationally. In this problem of optimization, we need to optimize between accuracy and computational speed by modulating the frequency of calculating the inverse in simplex iterations. To sum up, there are two interconnected issues in the revised simples: (1) numerical stability, and (2) computational speed. Computing the inverse $\mathbf{B^{-1}}$ from the original data is more accurate, but more memory and time costly.

IX. References

Bapi Chatterjee (2022). Revised Simplex Method. (https://www.mathworks.com/matlabcentral/fileexchange/26554-revised-simplex-method), MATLAB Central File Exchange. Retrieved November 26, 2022.

Desmos | Graphing Calculator. (n.d.). Desmos. https://www.desmos.com/calculator

HUA, W. (1998). *Application of the Revised Simplex Method to the Farm Planning Model* [MA Thesis]. Graduate College of the Oklahoma State University.

Ikpang, I. N., Tim, I. M., George, E. U., Okon, E. O., & Thompson, D. E. (2021). Application Of Revised Simplex Method For Profit Maximization In A Paint Company. *Researchjournali's Journal of Mathematics*, *8*(2). https://researchjournali.com/pdf/5826.pdf

Morgan, S. S. (1997). *A Comparison of Simplex Method Algorithms* [MA Thesis]. the Graduate School of the University of Florida.

Nasseri, S., Attari, H., & Ebrahimnejad, A. (2012). Revised simplex method and its application for solving fuzzy linear programming problems. *European J. of Industrial Engineering*, *6*(3), 259. https://doi.org/10.1504/ejie.2012.046670

Revised Simplex Method. (n.d.). BYJU'S. https://byjus.com/maths/revised-simplex-method/

Taha, H. A. (2017). *Operations Research: An Introduction, Global Edition* (10th ed.). Pearson.

Wolkowicz, H. (2005). *CO350 Linear Programming Chapter 9: The Revised Simplex Method*. University of Waterloo. https://www.math.uwaterloo.ca/~hwolkowi//henry/teaching/f05/350.f05/L27