Report 1

Revised Simplex Method

-

# Waleed Mustafa Awad

201501346

-

Math 404 Course:

Linear and NonLinear Programming

# Motivation:

The simplex method, originally developed by Dantzig, provides a solution to a linear programming problem; that is: an optimization (minimization/maximization) problem whose objective function and constraints (inequality/equality) are linear in the design variables. The method mainly depends on a table called Simplex Tableau, on which an iterative procedure is applied.

Some of the inefficiencies of the simplex algorithm for linear programming problems are later treated by the revised simplex method, which performs necessary updates only and stores necessary variables only.

# Theory:

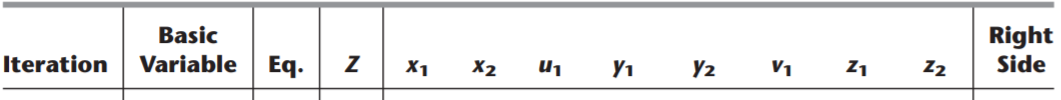
The standard form for a linear programming problem is as follows:

**Minimize F = cx**

**S.t. Ax b**

**All x 0**

Given an input problem is formulated as such, the simplex method proceeds to solve the problem, starting from a feasible point, by constructing what is known as a simplex tableau:



**Fig1. Simplex Tableau**

If the problem has **m** constraints and **n** variables then:.

* The basic variables column lists the indices of the **m** basic variables: variables having a non-zero value at the current iteration.
* The **Z** column lists the value of the function
* The next column lists the coefficients of the variables in the inequalities. Initially= **A**
* The **right side** column lists the right side of the equations (initially=**b**)

The simplex method follows a process of Gauss-Jordan elimination each iteration till a certain optimality criterion is satisfied. The pivot element (row, column) is decided on two basis:

1. The column of the non-basic variable with the most negative coefficient in the objective function (called entering variable)
2. The row that has the basic variable index in **Eq. column** with the smallest non-negative ration between **Right Side** column and its coefficients (called leaving variable)

Where the optimality criterion is that all the coefficients of the non-basic variables in the objective function are positive.

The revised simplex method argues the following:

We do not need to keep a full tableau and update it each iteration, we only need to keep:

1. Coefficients of the non-basic variables in **c**
2. Coefficients of the entering non-basic variable in **A**
3. Vector **b** (to calculate the right side)

# Algorithm:

The Algorithm of the revised simplex method then goes as follows (Lieberman):

1. **Initialization:** *Same as for the original simplex method.*

2. **While** the solution is not optimal (coefficients of non-basic variables in the objective function are all positive), **do**:

**Step 1** Determine the **entering variable**: Same as for the original simplex method. **Step 2** Determine the **leaving basic variable**: Same as for the original simplex method, except calculate only the numbers required to do this [the coefficients of the entering basic variable in every equation, and then, for each strictly positive coefficient, the right-hand side of that equation].1

**Step 3** Determine the new basic feasible solution (to be explained below)

3. **Optimality test**: Same as for the original simplex method, except calculate only the numbers required to do this test, i.e., the coefficients of the nonbasic variables in the objective function

Calculation of the new basic feasible solution is done as follows:

If **B** is an **m\*m** (square) matrix that contains the coefficients of the **m** basic feasible solutions in the current step, then we can set the values for the non-basic variables (stored in column vector **xB**) as:

**xB= .** This is true regardless of the step we are in.

Therefore, we only need to keep track of the basic variables.

# Implementation:

The revised simplex function is implemented in the attache MATLAB file (***Simplex.m***).

The function receives a matlab **struct called lin\_problem** that has all the problem parameters.

It has the elements **lin\_problem.A , lin\_problem.c, lin\_problem.b** corresponding to **A,c,b** in the Linear Programming problem under inspection.

Slack variables if size=m are then augmented to the matrix **A.**

1. **Initialize:**

|  |
| --- |
| xB=xs; %Init xB   B=A(:,xs'); %init B  Where **xs** contains the indices of slack variables, and **xB** contains the indices of the basic variables.   1. **While(Optimality not met) do:**   **Calculate a step:**    xvals(xB)= Binv\*b;  cB= c(xB);  z= cB\*xvals(xB);  Where **xvals** stores the values of the variables (n\*1 column vector). **cB** stores the coefficients of the basic variables in the objective function. **Z** stores the objective function value.  **Get entering variable:**  The index for the entering variable is found and retrieved in **c\_entering\_ind**  by retrieving the non-basic variable (i.e. variable with index not in **xB**) that has the most negative coefficient in **c** (objective function coefficients vector, dimensions = 1\***m**)  **Get leaving variable:**  The leaving variable is then retrieved by finding the minimum non-negative ratio of **xvals** over entering variable coefficients in **B**.   1. **The function ends when an optimal state is reached or when** |

# Application:

The function can be used similar only with inequality constraints, similar to the example:

Maximize **F**= 3**x1** + 5**x2**

**such that:**

**x1 <= 4**

**x2 <=12**

**3x1+2x2 <=18**

**Code:**

lin\_problem.maximize=true;   
 lin\_problem.c=[3,5];   
 lin\_problem.A=[1,0;0,2;3,2];   
 lin\_problem.b=[4,12,18]';

[x\_star, f\_star, num\_iterations, history] = ….

Simplex(lin\_problem);]

# Comparison with Matlab built-in function:

Comparing the aforementioned problem solution in MATLAB and in my function, we get the results:

==============================

Matlab function:

time= 0.015625 seconds

x\_star=[2;6]

f\_star=36.000000

==============================

My function:

time= 0.015625 seconds

x\_star=[2;6]

f\_star=36.000000

MATLAB solves using the dual simplex method while my solution utilizes the revised simplex method.