Optimization Project Report:

Simplex Method: Tableau Implementation

Connor Pitchford

Denzel Williams

December 6, 2018

1 Motivating the Project

The simplex method is the first algorithm we learned to solve linear programming problems without graphing the constraints. The simplex method has the power of iterating through the basic feasible solutions of the problem while checking each solution for optimality. Since its development in the 1940s, the simplex method is, arguably, the fastest and most efficient method for solving linear programming problems. With most solutions found in polynomial time, the simplex method has earned the medal of being a "fast" algorithm.

Although the method has proven to be efficient in solving problems, performing it by hand is unnecessarily tedious. Calculating and keeping track of the multiple matrices the method uses is tiring and leaves a lot of room for error. This project looks to explore a variation of the Simplex Method to better organize the data and thus speed up the time it takes to manually perform the algorithm as well as reduce the room for error.

The method used to implement the Simplex Algorithm uses tables. The goal of this project is to discuss and present how to use the Simplex Algorithm via tables. As usual in math, and what makes mathematics beautiful, there is more than one way to approach, and solve, a problem. That said, there are many types of Simplex Method tables. Some tables have minor discrepancies and others look completely different. Regardless of the physical appearance of the table, each implementation still captures and conserves the essence of the Simplex Method.

As said earlier, this project is looking to perform the Simplex Algorithm by hand and reduce the effort. Therefore, a table requiring the least amount of effort was required. The tableau implementation used is completely stripped down of excess elements and requires the least amount of restructuring.

2 Constructing the Initial Table

Beginning with a linear program in standard form, the first step is constructing the table. Like the simplex method, constructing the table requires writing the problem in canonical form, choosing an initial basis matrix, taking its inverse, and extracting a partition of the objective function coefficient vector, where the only values are the coefficients of the variables in the basis.

The problem used in this project:

$$\min\ z = x_1 + 5x_2 - 2x_3$$

$$\begin{aligned} s.t. \quad & x_1 + x_2 + x_3 + x_4 &&= 4 \\ & x_1 + x_5 &&= 2 \\ & x_3 + x_6 &&= 3 \\ & 3x_2 + x_3 + x_7 &&= 6 \\ & x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6, x_7 \geq 0 \end{aligned}$$

The initial basis chosen is composed of the variables x_1, x_3, x_6, x_7. Therefore, the initial matrix B is composed of the 1st, 3rd, 6th, and 7th column of the A matrix; in that order:

$$B = [A_1\ A_3\ A_6\ A_7],$$

where A is given by the matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 3 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The inverse of B is

$$B^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix}$$

Finally, the last matrix needed is a partition of the objective function vector, using only the values corresponding to the basis variables:

$$c_B^T = [c_1^T\ c_3^T\ c_6^T\ c_7^T]$$

where c^T is given by the matrix:

$$c^T = \begin{bmatrix} 1 & 5 & -2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.1 Table Quadrant Definitions

With the matrix definitions above, the problems with doing the Simplex Method by hand are apparent. Every iteration through the algorithm, one has to reconstruct the basis matrix, calculate its inverse, and repartition the objective function vector. With the usage of tables, these calculations only have to be performed once.

Using these matrices the table is pieced together in four separate parts. Each part having significance in reference to the Simplex Algorithm.

- **Upper Left Quadrant** - The objective function value, which is written in matrix form as $-c_B^T B^{-1} b$. *Referring back to the canonical form* $c^T x = -c_B^T x_b^* = c_B^T B^{-1} b$.
- **Lower Left Quadrant** - The current basic feasible solution given by $B^{-1}b$
- **Upper Right Quadrant** - The reduced costs given by $c^T - c_B^T B^{-1} A$. In the Simplex Method, optimality was checked by looking at the reduced cost of only the nonbasic variables. Otimality is reached when there are no negative values. In the table method, the reduced cost is checked for all variables. The reduced cost of basic variables being zero plays an important role in the table.
- **Lower Right Quadrant** - The matrix of direction vectors given by $B^{-1}A$. Each column in the bottom right portion of the table represents which values are used in the ratio test if the corresponding variable is entering the basis.

Compiling the information from the list the table initial table is formed as follows:

$-c_B^T B^{-1} b$	$c^T - c_B^T B^{-1} A$
$B^{-1}b$	$B^{-1}A$

For the example, the initial table is given by:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	2	0	7	0	2	-3	0	0
x_1	2	1	0	0	0	1	0	0
x_3	2	0	1	1	1	-1	0	0
x_6	1	0	-1	0	-1	1	1	0
x_7	4	0	2	0	-1	1	0	1

For readability the row and column headings were added to the table. Each column represents each variable in ascending order. This column order does not change throughout the entire process. The row headings change every iteration, as each row represents a current basic solution variable.

2.2 Extracting Meaning from the Table

Although the column and row headings are provided, it is important to understand the table without the headings.

2	0	7	0	2	-3	0	0
2	1	0	0	0	1	0	0
2	0	1	1	1	-1	0	0
1	0	-1	0	-1	1	1	0
4	0	2	0	-1	1	0	1

The first question to answer is, without headings, how does one tell which variables are in the basis. The answer is in the upper right quadrant. This quadrant represents the reduced costs. Variables in the basis always have a reduced cost of 0. Therefore every column with a reduced cost, an upper right quadrant, value of 0 corresponds to a basic variable i.e. x_1, x_3, x_6, and x_7 are all basic variables.

A slightly more challenging question is which row corresponds to which basic variable. This question is important as the order, with respect to the variables, of the basic solution will change. This ordering needs to be conserved or the algorithm will be damaged; and this ordering is not always ascending.

To figure out the row corresponding to the basic variable the process is the same. First get to a basic variable column, indicated by the zero in the reduced cost. Then follow the column down until you reach the 1. The row the 1 is in, is the row corresponding to the basic variable. For example, with the x_6 basic variable, the 1 in that column is in the 3rd row, therefore the 3rd row is the x_6 variable.

A question then arises, why, in a basic variable column, there are only zeros and one value of 1. The answer lies in the meaning of the lower right quadrant. As stated above the lower right quadrant is the matrix of direction vectors used in the ratio test. Assuming x_6 is entering the basis, each value in that column is used as a denominator in the ratio test. However, x_6 is already in the basis meaning the only candidate for replacing x_6 should be x_6

2.3 The Iteration Process

Determining the optimality of the linear program in the Tableau Implementation is the same as in the Simplex Method: the reduced cost is checked for a negative value. In the reduced cost quadrant of the example, there is a negative value. Therefore, the current solution $x = [2\ 0\ 2\ 0\ 0\ 1\ 4]$ is not optimal. The negative value is in the x_5 column, which means that x_5 is entering the basis. In the case there are more than one negative value, the entering variable is the variable with the most negative reduced cost.

2.3.1 The Ratio Test

As with the Simplex Method, a ratio test is done every iteration, to determine which variable is removed from the basis. The ratio test is exactly the same with the same conditions and requirements. The values in the x_5 column are used because the column represents $B^{-1}A_5$. Any negative value or zero in the column is not used in the test. The -1 value in the 3rd row is not used, meaning x_3 is not a candidate for removal. Additionally, all the accepted values in the basic solution value quadrant (lower left) are used. This gives the ratios:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	2	0	7	0	2	-3	0	0
x_1	(2)	1	0	0	0	(1)	0	0
x_3	2	0	1	1	1	(-1)	0	0
x_6	(1)	0	-1	0	-1	(1)	1	0
x_7	(4)	0	2	0	-1	(1)	0	1

$$\{\frac{2}{1}, \frac{1}{1}, \frac{4}{1}\}$$

Of these three numbers, the smallest ratio is used. 1 is the smallest ratio and it corresponds to the x_6 variable, meaning x_6 is leaving the basis and x_5 is taking its place.

2.3.2 Updating the Table Using Elementary Row Operations

How the basis is updated is the largest deviation from the standard Simplex Method. Instead of reforming the basis and doing all the calculations associated with the reforming, the table is directly updated using linear algebra techniques. The first step is identifying what is known as the pivot element. The pivot element is the number in the entering variable column and the exiting variable row. In the case of the example problem, the pivot element is in the x_5 column and in the x_6 row.

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	2	0	7	0	2	-3	0	0
x_1	2	1	0	0	0	1	0	0
x_3	2	0	1	1	1	-1	0	0
x_6	1	0	-1	0	-1	(1)	1	0
x_7	4	0	2	0	-1	1	0	1

Our objective is to remove x_6 and enter in x_5. Therefore, the x_5 row needs to meet the required conditions for a basic variable.

- The pivot element is equal to 1
- All other elements in the column (including reduced costs) are equal to 0

With these conditions met, x_5 is in the basis. For clarity, each row is defined below:

R_0	2	0	7	0	2	−3	0	0
R_1	2	1	0	0	0	1	0	0
R_2	2	0	1	1	1	−1	0	0
R_3	1	0	−1	0	−1	(1)	1	0
R_4	4	0	2	0	−1	1	0	1

To achieve the objective, elementary row operations are used on each row to first make the pivot element 1 and then to make each element in the pivot column 0. The row operations to achieve this objective are written below:

$$\frac{R_3}{\textit{pivot element}} \to R_3$$

$$3R_3 + R_0 \to R_0$$

$$-R_3 + R_1 \to R_1$$

$$R_3 + R_2 \to R_2$$

$$R_3 \to R_3$$

$$-R_3 + R_4 \to R_4$$

After the row operations, the new table is

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	5	0	4	0	−1	0	3	0
x_1	1	1	1	0	1	0	−1	0
x_3	3	0	0	1	0	0	1	0
x_5	1	0	−1	0	−1	1	1	0
x_7	3	0	3	0	0	0	−1	1

2.4 Repeating the Iteration Process Until Optimality

Once again, there is a negative in the reduced cost; therefore, the current solution $x = [1\ 0\ 3\ 0\ 1\ 0\ 3]$ is not optimal. The same methods are performed again and are performed until optimality is reached. x_4 is the entering variable. The ratio test is performed again. The only candidate for removal is x_1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	5	0	4	0	-1	0	3	0
x_1	1	1	1	0	1	0	-1	0
x_3	3	0	0	1	0	0	1	0
x_5	1	0	-1	0	-1	1	1	0
x_7	3	0	3	0	0	0	-1	1

The pivot element is the 1 that is in the x_4 column and the x_1 row.

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	5	0	4	0	-1	0	3	0
x_1	1	1	1	0	1	0	-1	0
x_3	3	0	0	1	0	0	1	0
x_5	1	0	-1	0	-1	1	1	0
x_7	3	0	3	0	0	0	-1	1

Using linear algebra techniques, the pivot element is scaled to 1 and every element in its column becomes a 0. Defining the rows the same as before, the elementary row operations performed are:

R_0	5	0	4	0	-1	0	3	0
R_1	1	1	1	0	1	0	-1	0
R_2	3	0	0	1	0	0	1	0
R_3	1	0	-1	0	-1	1	1	0
R_4	3	0	3	0	0	0	-1	1

$$\frac{R_1}{\textit{pivot element}} \to R_1$$

$$R_1 + R_0 \to R_0$$

$$R_1 \to R_1$$

$$R_2 \to R_2$$

$$R_1 + R_3 \to R_3$$

$$R_4 \to R_4$$

The new table with x_4 in the basis and x_1 removed is given by:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	6	1	5	0	0	0	2	0
x_4	1	1	1	0	1	0	-1	0
x_3	3	0	0	1	0	0	1	0
x_5	2	1	0	0	0	1	0	0
x_7	3	0	3	0	0	0	-1	1

There are no negative values in the reduced cost quadrant, meaning optimality has been reached. The current solution $x = [0\ 0\ 3\ 1\ 2\ 0\ 3]$ is optimal.

3 Computation vs. Expectation

As expected, the Tableau Implementation is more efficient than the standard Simplex Method when performed by hand. The reason using the table is more efficient is it does not require the two most computational heavy parts of the Simplex Method. The Tableau Implementation doesn't require the re-partitioning of matrices every iteration and it does not require the heaviest computation, finding the inverse of the basis matrix. Which can be calculated using either Guass-Jordan Elimination or the use of Adjugate matrix, which requires transposing the cofactor matrix and finding the determinant of the basis matrix. Even with a simple 2×2 matrix, calculating the inverse is a heavy computational and time consuming task.

4 Future Implementations

The way we send and receive things has drastically changed. From only word of mouth through physical interaction taking months at a time, to now being able to send our words across the globe in a blink of an eye. Sending and receiving objects isn't as instantaneous as sending words, but we can send objects with little ease and in a short amount of time. Millions of packages are sent on a daily basis, and making sure that they get to the right place, in the least amount of time, with the least amount of effort is a giant optimization problem. Every mechanical vehicle known to man is used in to solve these problems; planes, boats, trucks, cars, and even people. Now there is a new form of delivery being developed and a new optimization problem to be solved. Package delivery via drones is a hot topic and with it a new network problem to be solved. If drone delivery gets approved, then there will be a new network problem to be optimized.

In mathematics, developing the intuition to solve complex problems first begins with the solution to a simpler problem. Although optimizing the drone network problem is not likely to be solved by hand, solving a simplified problem could be solved using the using the Tableau Implementation of the Simplex Method. The reason it would be done by hand is because it wouldn't require the usage of complex and expensive software. The Tableau Implementation of the Simplex Method may not be the solution to an important problem, but it can be used as a step in the right direction.

Appendix: Special Cases

Degeneracy: a linear problem is degenerate if in a basic feasible solution, one of the basic variables takes a zero value. Degeneracy in a linear program means there are more than one non-optimal solutions which produce the same solutions/objective value. In some cases of degeneracy, there is a chance the linear program will cycle. Meaning even after the algorithm moves from extreme point to extreme point, the solution never changes. Degeneracy manifests itself in a table when there is a tie for departing variables. An example of a table formed from a degenerate program is shown below:

			x_1	x_2	x_3	x_4	x_5
R_0		0	-2	-1	0	0	0
R_1	x_3	12	4	3	1	0	0
R_2	x_4	8	4	1	0	1	0
R_3	x_5	8	4	2	0	0	1

With x_1 as the entering variable, there is a tie in Row 2 and Row 3. After using either as a Pivot Element, the Tableau will degenerate, as follows:

			x_1	x_2	x_3	x_4	x_5
R_0		4	0	$\frac{-1}{2}$	0	$\frac{1}{2}$	0
R_1	x_3	4	0	2	1	-1	0
R_2	x_1	2	1	$\frac{1}{4}$	0	$\frac{1}{4}$	0
R_3	x_5	0	0	1	0	-1	1

Alternate Optima: Alternate optima is similar to degeneracy. An alternate optima occurs when more than one extreme point produces the optimal solution. Alternate optima manifests itself when the reduced cost of a nonbasic variable is zero at optimality. The reason this shows alternate optima is because the nonbasic variable with reduced cost zero were to enter the basis, there would be no need to perform a row operation on R_0 and the objective function value remains the same although the solution changed. An example of a alternate optima linear problem is shown below:

			x_1	x_2	x_3	x_4
R_0		42	0	0	2	0
R_1	x_2	3	$\frac{2}{7}$	1	$\frac{1}{7}$	0
R_2	x_4	15	$\frac{45}{7}$	0	$\frac{-2}{7}$	1

The nonbasic x_1 has a zero in the reduced cost. By letting x_1 into the basis, it is observed the objective value won't change.

			x_1	x_2	x_3	x_4
R_0		42	0	0	2	0
R_1	x_2	$\frac{7}{3}$	0	1	$\frac{7}{45}$	$\frac{-2}{45}$
R_2	x_1	$\frac{7}{3}$	1	0	$\frac{-2}{45}$	$\frac{7}{45}$

Unbounded: A linear program is unbounded when the objective value can be made as large or small as one wishes. Unboundedness occurs in an non-optimal table when the ratios for the entering variable are all negative or infinity (zero). An example of a linear program that has an unbounded table is shown below:

			x_1	x_2	x_3	x_4
R_0		80	0	0	-4	3
R_1	x_1	30	1	0	-1	1
R_2	x_2	20	0	1	-2	1

x_2 is our entering variable, its column is all negative numbers and thus negative ratios. The variable x_2 is thus unbounded. Having an unbounded variable means the tableau is unbounded.