Linear Programming and Simplex Algorithm

Phineas Pham

April 2023

1 Introduction:

Algorithms are used in many real-world problems. To solve optimization problems, linear programming is among the most common techniques. It is often used in business, engineering, economics, and other fields to help make decisions that optimize resources and achieve the highest outcome within the context constraints. There are many applications of linear programming, as the field is still growing in recent years. From optimizing the model for ingredient shopping for a soup kitchen, to finding the optimal combination of oil making that saves oil companies millions of dollars every year, linear programming is used in a wide range of problems.

Linear programming (LP), also called as linear optimization, is a mathematical optimization technique used to find the best solution to a problem whose requirements are represented by linear relationships. The goal of linear programming is to find the values of the variables that will maximize or minimize the objective function while satisfying all of the model's constraints. This is typically done using a technique called the simplex method, which involves iteratively improving a solution until it becomes optimal.

The simplex algorithm is a powerful algorithm that can efficiently solve linear programming problems with thousands or even millions of variables and constraints. In short, it is an iterative algorithm that starts with a feasible solution and gradually improves it until an optimal solution is found.

In this paper, we will walk through the process of formulating a mathematical model for LP problem. Then, we will learn Simplex algorithm and its application in solving these LP problems. We will walk through an example of Simplex algorithm first, and then we formally define the algorithm. Later in the paper, we will walk through some of the recent research on innovative applications of linear programming in different industries and scenarios.

2 Terminologies:

1. Multivariate optimization:
   A multivariate function contains terms each of which is composed of multiple variables, continuous variable raised to (and only to) the power of 1. For example: $f(x) = 3x + 7y + 1$.

2. Objective Function:
   It is a linear function of the decision variables expressing the objective of the decision-maker. The most typical forms of objective functions are: maximize $f(x)$ or minimize $f(x)$.

3. Decision Variables:
   Decision variables in linear programming represent decisions that needs to be made to solve a problem. They are used to represent the values of the quantities to be optimized. The values of these variables are determined by the optimization algorithm used to solve the problem. Decision variables help to formulate the constraints and objective function of the problem. For example, in solving function $f(x) = 3x + 7y + 1$, our decision variables are x and y.

4. Constraints:
   These are equations arising out of practical limitations. We can have constraints for decision variables. For the function $f(x)$ above, we can put some constraints for x, y, or a combination of x and y:

$$x \geq 5,\ y \leq 2;$$
$$x + y \geq 3.$$

5. Non-negativity Restrictions:
   In most practical problems, the variables are required to be non-negative: $x_j \geq 0$, for $j = 1.....n$.
   This constraint is called a non-negativity restriction. Sometimes variables are required to be non-positive, or in fact, may be unrestricted.

6. Feasible Solution:
   Any non-negative solution which satisfies all the constraints is known as a feasible solution. The region comprising all feasible solutions is referred to as feasible region. Here is an example of how a feasible region looks like in LP:

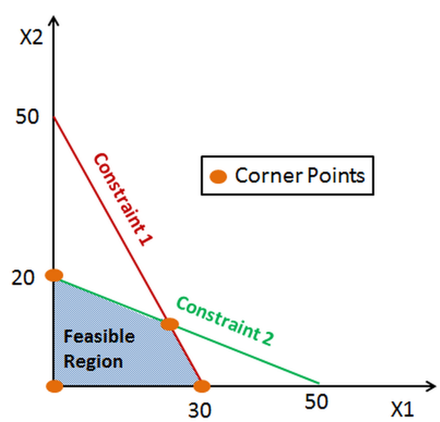

Figure 1: Feasible Region (Linear Programming, 2023)

   We have constraints 1 and 2 with non-negative variables X1 and X2. Thus, our feasible region, or solution space, is the blue shaded area.

7. Optimal Solution: The solution where the objective function is maximized or minimized is known as optimal solution.

3 Problem definition:

In linear programming, we start creating the model by defining decision variables, then add a set of linear equations and inequalities that represent the constraints or limitations of the problem. We can also have an objective function that we want to maximize or minimize, such as profit or cost. The solution of a LP problem reduces to finding the optimum value (largest or smallest, depending on the problem) of the linear expression (called the objective function).

For example, we try to create a mathematical model to make decisions on a selection of ingredients to shop that minimize the cost but meet the constraints of the amount of protein. We can solve the problem by setting up a mathematical model using linear programming. For every ingredients i, we have c_i as the associated cost ($ per gram) and x_i as the amount to shop (in grams). Our minimizing function f is as follows:

$$\min f(x_1, ..., x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n;$$

subject to a set of constraints expressed as inequalities:

$$c_1x_1 \geq 10$$

(Spend at least $10 on meat - item 1)

$$\sum_{i=2}^{5} c_ix_i \geq 20$$

(Spend at least $20 on vegetables - items 2 to 5)

$$\vdots$$

$$x_1, x_2, ..., x_i \geq 0.$$

Then, our job is to find the optimal value of f where all the constraints are satisfied.

4 Simplex algorithm:

4.1 Motivation and Terminologies:

The Simplex method was invented by Dantzig (first published in 1951) to solve any linear programming problem. Along other algorithms like Barrier, interior-point, or First-order methods, Simplex algorithm was the first practical LP algorithm and remains the most popular.

Before performing Simplex algorithm, we have to manipulate the model into the standard form: A linear programming problem that appears in a particular form where all constraints are equations and all variables are non-negative and are said to be in standard form. There should be Slack and surplus variables, Basic and non-basic variables, Admissible solution.

Slack and Surplus variable: Slack and surplus variables are variable used to transform an **inequality** constraint into an **equality** one.

a. Constraints of type ($\leq$) : for each constraint i of this type, we add a slack variable e_i, such that e_i is non-negative. Example: $x_1 + 2x_2 \leq 2$ translates into $x_1 + 2x_2 + e_1 = 2$, $e_1 \geq 0$.

b. Constraints of type ($\geq$) : for each constraint i of this type, we add a surplus variable e_i, such that e_i is non-negative. Example: $x_1 + 2x_2 \geq 2$ translates into $x_1 + 2x_2 - e_2 = 2$, $e_2 \geq 0$.

Basic and non-basic variables: Consider a system of equations with n variables and m equations where $n \leq m$. A basic solution for this system is obtained in the following way:

a. Set $(m+1)^{th}, (m+2)^{th}, ..., n^{th}$ variables equal to zero. We call these non-basic variables.

b. Solve the system for the m remaining variables. These variables are called basic variables.

c. The vector of variables obtained is called the basic solution (it contains both basic and non-basic variables).

Admissible solution: Each basic solution of an LP problem for which all variables are non-negative, is called an admissible basic solution. This admissible basic solution corresponds to an extreme point (corner solution).

Tableau Method for Simplex: The tableau method is meant to be a shorthand way of describing the operations of the Simplex method. The general ideas of the tableau method are as follows:

- All variables will be on the left hand side, and the values of the equations on the right hand side (RHS).
- As with any Simplex method, there will be one basic variable per row. The column of any basic variable will have all zeros, except for its rows, which will have a one.
- Only the coefficients of the variables will be stored in the tableau.

Thus, here is an example of a tableau:

$z\ x_1\ x_2\ \dots\ x_n$	RHS	Basic Variables
COEFFICIENTS	values of RHS	Basic variables
...		

4.2 Methodologies:

Now, we look into how Simplex Algorithm works. Consider the following expression as the general linear programming problem, which we will transform into standard form and use Simplex algorithm to solve it:

$$max \sum_{i=1}^{n} c_i * x_i$$

With the following constraints:

$$\text{s.t.} \ \sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{ for } i = 1, 2, ..., m$$

$$x_j \geq 0 \text{ for } j = 1, 2, ..., n$$

First, we add symbol ϕ which represent the objective functions and slack variables z_i (to transform ($\leq$) inequality constraints into equality constraints):

$$\phi = \sum_{i=1}^{n} c_i * x_i$$

$$z_i = b_i - \sum_{j=1}^{n} a_{ij} * x_j \text{ for i = 1, 2, ..., m}$$

The new slack variables z_i may get confused with original values x_i. Thus, we append those slack variables z_i to the end of the list of x_i variables with the following expression:

$$\phi = \sum_{i=1}^{n} c_i * x_i$$

$$x_{n+1} = b_i - \sum_{j=1}^{n} a_{ij} x_{ij} \text{ for i = 1, 2, ..., m}$$

We call the 2 equations above our starting dictionary (ϕ and x_i). As the simplex method progress, the starting dictionary switches between the dictionary while seeking for optimal values. Every dictionary will have m basic variables which form the feasible area, as well as n non-basic variables which compose the objective function. We rewrite the formula with values with bars, suggesting if they will change accordingly with the progression of the simplex method:

$$\phi = \bar{\phi} + \sum_{i=1}^{n} \bar{c}_i * x_i$$

$$x_i = \bar{b_i} - \sum_{j=1}^{n} \bar{a_{ij}} * x_{ij} \text{ for i = 1, 2, ..., m}$$

4.3 Numerical example:

We now go through an example to gain better understanding how the Simplex algorithm works. After showing the steps of the Simplex algorithm, we later give a formal definition in Section 4.4. We have a linear programming problem as follows:

$$max\ 4x_1 + x_2 + 4x_3$$

with the following constraints:

$$2x_1 + x_2 + x_3 \leq 2$$

$$x_1 + 2x_2 + 3x_3 \leq 4$$

$$2x_1 + 2x_2 + x_3 \leq 8$$

$$x_1, x_2, x_3 \geq 0$$

We will solve the linear programming problem by using Simplex algorithm. To use Simplex algorithm, we need to transform the LP problem into standard form. Our first step is to add slack variables:

$$z - 4x_1 - x_2 - 4x_3 = 0$$
$$2x_1 + x_2 + x_3 + s_1 = 2$$
$$x_1 + 2x_2 + 3x_3 + s_2 = 4$$
$$2x_1 + 2x_2 + x_3 + s_3 = 8$$
$$x_1, x_2, x_3, s_1, s_2, s_3 \geq 0$$

Then, we can derive simplex tableau:

x_1	x_2	x_3	s_1	s_2	s_3	z	b
2	1	1	1	0	0	0	2
1	2	3	0	1	0	0	4
2	2	1	0	0	1	0	8
-4	-1	-4	0	0	0	1	0

In the last row, the column with the smallest value should be selected. Although there are two smallest values (-4), the result will be the same no matter of which one is selected first. For this solution, the first column is selected. After the least coefficient is found, the pivot process will be conducted by searching for the coefficient $\frac{b_i}{x_1}$. Since the coefficient in the first row is 1 and 4 for the third row, the first row should be pivoted. And following tableau can be created: By continuing

x_1	x_2	x_3	s_1	s_2	s_3	z	b
1	0.5	0.5	0.5	0	0	0	1
1	2	3	0	1	0	0	4
2	2	1	0	0	1	0	8
-4	-1	-4	0	0	0	1	0

the row operation for every other rows (other than first row) in column 1 until x_1 values are zeroes, we get this tableau: Because there is one negative value in last row, the same processes should be

x_1	x_2	x_3	s_1	s_2	s_3	z	b
1	0.5	0.5	0.5	0	0	0	1
0	1.5	2.5	-0.5	1	0	0	3
0	1	0	-1	0	1	0	6
0	1	-2	2	0	0	1	4

performed again. The smallest value in the last row is in the third column. And in the third column, the second row has the smallest coefficients of $\frac{b_i}{x_3}$ which is 1.2. Thus, the second row will be selected for pivoting. The simplex tableau is the following:

x_1	x_2	x_3	s_1	s_2	s_3	z	b
1	0.2	0	0.6	-0.2	0	0	0.4
0	0.6	1	-0.2	0.4	0	0	1.2
0	-0.1	0	0.2	0.6	-1	0	-4.2
0	2.2	0	1.6	0.8	0	1	6.4

There is no need to further conduct calculation since all values in the last row are non-negative. From the tableau above, x_1. x_2, x_3 and z are basic variables since all rows in their columns are 0's except one row is 1. Therefore, the optimal solution will be $x_1 = 0.4, x_2 = 0, x_3 = 1.2$, achieving the maximum value: $z = 6.4$.

4.4 Simplex algorithm:

First, we define the input and output of the Simplex:

Input:
For Simplex algorithm to work, we first need to make the linear programming problems into standard form like the simplex tableau from Numerical Example above. Then, we have: a set of decision variables (such as $x_1, x_2, \ldots, x_n$) that can take on real values and a set of linear inequalities in the form of constraints (such as Ax ≤ b). We input these as variables to simplex()

c: coefficients of decision variables from Objective function
A: list of coefficients of decision variables from list of constraints
b: RHS values of list of constraints

Output:
Optimal solution and objective value to the LP problem.

Based on the numerical example, we can develop an informal steps of Simplex algorithm:

Algorithm 1 Simplex Algorithm

Initialize the Simplex tableau of the inputs.
while there exists a negative coefficient in the bottom row of the tableau **do**
    Select a pivot column with the most negative coefficient.
    Select a pivot row using the ratio test.
    Divide the pivot row by the pivot element.
    Use row operations to make the other entries in the pivot column zero.
Extract the solution from the tableau.

Here is the pseudocode of Simplex algorithm:

Algorithm 2 simplex()

tableau = initialTableau(c, A, b)
while canImprove(tableau) **do**
    pivot = findPivotIndex(tableau)
    pivotAbout(tableau, pivot)
return primalSolution(tableau), objectiveValue(tableau)

Let's define the functions used in simplex():

primalSolution(tableau) returns the coefficients of decision variables in the optimal solution.
objectiveValue(tableau) return the optimal objective value.
canImprove(tableau) check if there exists a negative coefficient in the bottom row of the tableau.
findPivotIndex(tableau) find the column with the most negative coefficient and the row of that column with the most negative coefficient.
pivotAbout(tableau, pivot), in short, modify the tableau by dividing the pivot row by the pivot element and using row operations to make the other entries in the pivot column zero. Here is the pseudocode for pivotAbout(tableau, pivot):

Algorithm 3 pivotAbout(tableau, pivot)

```
i, j = pivot //Unpack the pivot indices
pivotDenom = tableau[i][j] //Compute the pivot denominator

//Make the pivot element equal to 1
pivotRow = tableau[i]
for k = 0 to number of columns in tableau - 1 do
   pivotRow[k] = pivotRow[k] / pivotDenom
tableau[i] = pivotRow

//Eliminate other variables using the pivot row
for k = 0 to number of rows in tableau - 1 do
   if k != i then
      pivotCoeff = tableau[k][j]
      pivotRowMultiple = [pivotCoeff * x for x in pivotRow]
      for for l = 0 to number of columns in tableau - 1 do
         tableau[k][l] = tableau[k][l] - pivotRowMultiple[l]
```

Proof of correctness:

The proof of correctness for the Simplex algorithm is quite complex and involves a number of mathematical concepts, including linear algebra and optimization theory. The proof is based on the following key observations:

1. Any linear programming problem can be transformed into an equivalent form that has a feasible solution.
2. The feasible region of a linear programming problem is a convex polytope (Figure 1).
3. The optimal solution to a linear programming problem lies at one of the vertices of the feasible region.
4. The simplex algorithm moves from vertex to vertex along the edges of the feasible region, always moving in the direction of increasing objective function value (negative value if minimizing).
5. The algorithm terminates when it reaches an optimal solution.

Using these observations, it can be shown that the Simplex algorithm always produces the correct optimal solution if one exists, and terminates in a finite number of steps.

Runtime:

In worst case scenario, Simplex's worst run-time is $O(2^n)$ as it has to visit all 2^n vertices of the feasible region (Klee & Minty 1972), which could also be the same point being visited repeatedly if considering degeneration. The number of 2^n is based on the mathematical concept that there are possibly 2^n corners in the feasible region with n constraints. For example, in figure 1, the feasible region has 4 (2^2) corners as there are $n = 2$ constraints. Kelner and Spielman (2006) introduced a polynomial time randomized simplex algorithm that truly works on any inputs, even the bad ones for the original simplex algorithm.

5 Related Work:

The simplex method can be used in many programming problems since those will be converted to LP (Linear Programming) and solved by the simplex method. Besides the mathematical application, industrial planning use this method to maximize the profits or minimize the resources needed. Here, we will provide literature review on applications of linear programming on different topics and scenarios.

1. **Application of LP for Optimal Investments in Software Company:**
   Published by Mustafa, A. O., Sayegh, M. A., Rasheed, S. in 2021, the paper proposes an application of linear programming in helping young software companies make decisions on optimal investment for best return. The goal of the research is to derive a mathematical model using Linear Programming and the Simplex Algorithm to determine the number of projects in each category to be taken at one period under the pre-specified limitations to achieve maximum profit.

   First, they categorize software projects into four categories: Mobile Applications, Web Applications, Desktop Applications, and Enterprise Resource Planning (ERP) Applications. The numbers and the constraints are shown in Table 1:

Name	Project type				Maximum in 4 months
	Mobile App	Desktop App	Web App	ERP App	
Expert developers	1	4	2	7	18
Senior Developers	4	4	6	9	30
Software Testers	2	4	5	7	20
Servers	1	2	1	3	20
Storage (in GB)	250	925	500	6500	15,000
Fixed Expenses ($)	4700	5000	8000	14,000	45,000
Profit ($)	9000	12,000	17,000	35,000	-

Table 1: Data table from paper of Mustafa, A. O., Sayegh, M. A., Rasheed, S.

   With a number of constraints and background information, researchers were able to formulate a linear programming model:

Let x_1, x_2, x_3, x_4 be the number of mobile applications, desktop applications, web applications, and ERP systems to take. The LP model for the above resources, in Standard Form, is given by:

$$Z(max) = \$9,000x_1 + \$12,000x_2 + \$17,000x_3 + \$35,000x_4$$
$$+0s_1 + 0s_2 + 0s_3 + 0s_4 + 0s_5 + 0s_6 + 0s_7 + 0s_8$$

with the following constraints:

$$x_1 + 4x_2 + 2x_3 + 7x_4 + s_1 = 18$$
$$4x_1 + 4x_2 + 6x_3 + 9x_4 + s_2 = 30$$
$$2x_1 + 4x_2 + 5x_3 + 7x_4 + s_3 = 20$$
$$250x_1 + 925x_2 + 500x_3 + 6500x_4 + s_5 = 15000$$
$$4700x_1 + 5000x_2 + 8000x_3 + 14000x_4 + s_6 = 45000$$
$$x_4 + s_7 = 1$$
$$x_1 + s_8 = 3$$
$$x_1, x_2, x_3, x_4, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8 \geq 0$$

Solving the model with simplex algorithm, we get the optimal solution:

$$x_1 = 3, x_2 = 0, x_3 = 1.4, x_4 = 1, Z = 85800$$

With the result, researchers are able to suggest:

(a) The number of Mobile Applications to be taken in one period should be 3.

(b) The number of Web Applications to be taken in one period should be 0.

(c) The number of Desktop Applications to be taken in one period should be 1 or 2 since 1.4 is not an integer.

(d) The number of ERP Applications to be taken in one period should be 1.

2. **A review of applications of linear programming to optimize agricultural solutions:** In this research, Farrukh Nadeem used various LP applications including feed mix, crop pattern and rotation plan, irrigation water, and product transformation to enhance various facets of the agriculture sector.

3. **An application on family meal planning under covid-19 scarcity constraints:** Steven T. Joanis collected data on ingredients and nutrients requirement to build a LP model that optimizes the food requirement for a theoretical family with only one visit to supermarket a month.

6 Conclusion:

Linear programming and Simplex algorithm has made major contributions to today's decision making problems. By formulating the problem into mathematical models and transforming it into standard formula, we can use Simplex algorithm to figure out the optimal solution for the problem above. That leaves to the main challenge is to try to formulate the problem from real-world context into mathematical models with linear programming concepts. The more complex the problem is, the more constraints and variables the model has. Recently, companies have developed tools like Gurobi to solve these linear programming models, but the algorithms and technology built behind remains the company's secret sauce. Thus, Simplex algorithm becomes handy to solve the linear programming problems.

7 Reference:

1. Mustafa, A. O., Sayegh, M. A., Rasheed, S. (2021). Application of Linear Programming for Optimal Investments in Software Company. Open Journal of Applied Sciences, 11(10), 1092–1101. https://doi.org/10.4236/ojapps.2021.1110081

2. Alotaibi, A., Nadeem, F. (2021). A review of applications of linear programming to optimize agricultural solutions. International Journal of Information Engineering and Electronic Business, 13(2), 11–21. https://doi.org/10.5815/ijieeb.2021.02.02

3. T. Joanis, S. (2020). Family meal planning under covid-19 scarcity constraints: A linear programming approach. Journal of Food and Nutrition Research, 8(9), 484–495. https://doi.org/10.12691/jfnr-8-9-5

4. j2kun. (2018, April 23). Linear programming and the simplex algorithm. Math Programming, from `https://jeremykun.com/2014/12/01/linear-programming-and-the-simplex-algorithm/`

5. Encyclopedia Britannica, inc. (n.d.). Linear Programming. Encyclopedia Britannica, from `https://www.britannica.com/science/linear-programming-mathematics`

6. The steps of the simplex algorithm - hec montréal. (n.d.), from `https://www.hec.ca/en/cams/help/topics/The_steps_of_the_simplex_algorithm.pdf`

7. Linear programming terminology. (n.d.), from `http://www.universalteacherpublications.com/univ/ebooks/or/Ch2/bscterms.htm`

8. Simplex algorithm. Simplex algorithm - Cornell University Computational Optimization Open Textbook - Optimization Wiki, from `https://optimization.cbe.cornell.edu/index.php?title=Simplex_algorithm`

9. Klee, V. and Minty, G. (1972) How Good Is the Simplex Algorithm? In Inequalities. Academic Press, New York.

10. Kelner, J. A., amp; Spielman, D. A. (2011). Complexity of the simplex algorithm and polynomial-time algorithms. Linear Programming and Network Flows, 393–451. from `https://doi.org/10.1002/9780471703778.ch8`