Linear Programming

Arjun Kulkarni

July 12, 2021

Abstract

In this paper, we give a brief overview of linear programming. We discuss the theory behind linear programming and analyse algorithms that solve the linear programming problem. We also look at open problems in linear programming.

1 Introduction and Examples

First, we will begin by defining linear programming.

Definition 1. *Linear programming is an optimization tool to maximize or minimize a linear function in several variables under a set of linear constraints.*

More formally, we must find a vector $x \geq 0$ which maximizes $y^T x$ subject to $Ax \leq b$ where A is a matrix, and y is a vector. The function $f(x) = y^T x$ is known as the objective function. We will look at a simple example to better understand linear programming problems.

Example 1. *Maximize the function* $f(x, y) = 5x - 2y$ *subject to the constraints,* $0 \leq x \leq 6$, $0 \leq y \leq 6$, $y - x \leq 0$, *and* $3x - 2y \leq 6$.

In order to solve this problem, we would have to use algorithms discussed later in the article. But from this example, we can see that $y = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$, the objective function is $f(x, y) = 5x - 2y$, $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \\ 3 & -2 \end{bmatrix}$, and $b = \begin{bmatrix} 6 \\ 6 \\ 0 \\ 6 \end{bmatrix}$.

For this particular example, the optimal x vector would be $\begin{bmatrix} 6 \\ 6 \end{bmatrix}$. Now let us move on to some of the theory behind linear programming and a simple yet powerful method to solving linear programming problems known as the graphical method.

2 Theory

First, we will define terms to make discussion of linear programming theory easier.

Definition 2. *A hyperplane is the graph of the solution set of a linear equation.*

Definition 3. *A halfspace is the graph of the solution set of a linear inequality.*

From this, we can see that the solution set of a system of linear inequalities is the intersection of all halfspaces that represent these inequalities. We also have a name for the graph of this solution set.

Definition 4. *A polytope is a region in R^n bounded by finitely many hyperplanes.*

And this means that the solution set of a system of linear inequalities is a polytope.

Definition 5. *The facets of a polytope are defined as the intersection of the hyperplanes bounding the polytope.*

Examples of facets are vertices which are 0-dimensional facets, and edges which are 1-dimensional facets.

Definition 6. *A region S in R^n is convex, if for any $a, b \in S$, the line segment between a and b is contained in S.*

Convexity will be used a lot throughout the article because of the following lemma.

Lemma 1. *The intersection of halfspaces is convex.*

Proof. First, let us denote the intersection of halfspaces to be S. This region will be defined by $Ax \leq c$. For any two vectors a and b, we must find a way to represent all points on the line segment connecting a and b. It is not hard to see that we can represent each point as a function of $t \in [0, 1]$, as $a + t * (b - a) = a(1 - t) + b * t$. This means that $A(a(t - 1) + t * b) = (1 - t) * A * (a) + (t) * A * b \leq t * c + (1 - t) * c = c$ which means that all points on the line connecting a and b are part of S, which concludes the proof. □

Since the solution set for a system of linear inequalities is an intersection of halfspaces, using our lemma 1 we get the following theorem.

Theorem 1. *The region that represents the solution to a set of linear inequalities is a convex polytope.*

Now we will end on the graphical method for solving linear programming problems.

Theorem 2. *The values that maximize a linear objective function will always be located on the vertices of the solution set of the linear constraints.*

Proof. First, we must realize that if we have a line and a linear function, the minimum and maximum values of the function will take place on the endpoints of that line. The reason for this is the linear function will keep increasing or keep decreasing when you go from one point to another, so to decrease the value of the function as much as possible you must go to one endpoint, and to increase the value of the function as much as possible you must go to the other endpoint. Now take any point in our solution set and draw a line through it. Without loss of generality if we are maximizing the function, then one of the endpoints of the line will have make the linear function have a greater value. Next, we take the lowest dimensional facet that the endpoint is on, and we draw a line through the endpoint such that the line is contained in the facet. The endpoint of that line when plugged into the function will have at least as great of a value. We can keep doing this until we get to a 0-dimensional facet, or a vertex. This means that the vertices will generate values of the function that are at least as large as values generated by other points, and thus one of the vertices will maximize the linear function. □

And using this theorem, we find the graphical method to solving linear programming problems which involves plugging in all vertices of the solution set of our constraints into the linear function, and seeing which one maximizes or minimizes the function. Now we will look at algorithms trying to solve the linear programming problem.

3 Algorithms

First, we will go over the classical simplex algorithm which goes through vertices of our constraint solution set region, and constantly improves upon its optimal solution to get the real optimal solution.

Theorem 3. *An algorithm that solves the linear programming problem is known as the simplex algorithm, and its steps are as follows:*

1. *Write the objective function as a variable, and bring all variables to one side of the objective function equation.*

2. *Convert all linear inequalities into equations using slack variables.*

3. *Construct a matrix that represents the system of linear equations and put the objective function with the objective function equation in the bottom row.*

4. *Find the pivot column by seeing which coefficient in the bottom row is the smallest, and the column of that row is the pivot column.*

5. *Calculate coefficients of each row. The smallest coefficient will identify the row we will work with. The intersection of this row and the column in step 4 is known as the pivot element.*

6. *Use gaussian elimination to make all other entries in this column 0.*

7. *If there are negative entries in the bottom row, repeat from step 4. Otherwise, we are finished.*

8. *To find our answers, take all columns that don't consist of a 1 and all other entries to be 0's, and set their corresponding variables to* 0*. After this, solve the system of equation to get optimal values of all variables.*

We will not prove that this algorithm works, we will only provide brief explanation. For step 4, the reason we choose the most negative entry is because it has the largest coefficient in the objective function, and using it as a pivot variable will increase the objective function the quickest. The reason we choose the smallest quotient in step 5 is because the quotient identifies the constraint on our pivot variable, and using the row of the smallest quotient guarantees we do not violate our constraints. Finally, we are finished when there are no more negative entries in the bottom row, and this happens because the RHS of the bottom row will be as large as possible because all variables are positive, and this will identify the maximum value of the objective function by setting other variables to 0. As far as the runtime analysis, the algorithm has exponential runtime. This is because it visits 2^n vertices in the worst case scenario. However in practice, it runs in roughly polynomial time. There are only certain bad inputs that cause it to go through each vertex.

Now we will briefly go over interior point method. Interior point algorithms run in polynomial time and traverse the interior of the solution set region of the constraints, rather than through the vertices like the simplex algorithm. They can be used to solve nonconvex and convex optimization problems. However the theory behind the algorithms and the algorithms themselves are require background knowledge that is too deep for this article. This concludes our study of linear programming algorithms.

4 Applications

In this section we will discuss applications of linear programming and problems related to linear programming. First, we will discuss the following question.

Question 1. *Can adjusting pivot rules create simplex variants that are polynomial time?*

This question arises from the disproven Hirsch conjecture which discusses the diameter of polytopes in Euclidean space. So far, there has been no answer to this question and it remains an open problem.

Next we move onto the field of integer linear programming. Integer linear programming problems are where all unknown variables are integers. This problem is classified as NP-hard, and there are algorithms that have solved this problem like the Cutting-Plane Method and the Branch and Bound

Method. We will not go further into these algorithms because they are beyond the scope of this book.
Finally, we will be discussing flow networks. In flow network problems, we want to maximize the flow through our sink node and this can be formulated as maximizing a linear objective function. Because of conservation of flow, we will also have a set of linear constraints. And so we can use linear programming algorithms to solve maximum flow problems. This concludes our study of linear programming.

5 References

1. "Linear Programming." Brilliant Math amp; Science Wiki, brilliant.org/wiki/linear-programming/.

2. Libretexts. "4.2: Maximization By The Simplex Method." Mathematics LibreTexts