

# APPM3310 Project Proposal: Linear Programming

Darice Guittet and Benjamin Shoeman

3 November 2017

## 1 Developing Simplex Method Algorithm and Solving Shortest-Path-in-Network Problem

Linear programming is the process of optimizing a system by maximizing or minimizing an objective function given inequalities as constraints. Suppose we want to find the values of  $\mathbf{x} \geq 0$  that maximize:

$$c_1x_1 + \cdots + c_nx_n = \mathbf{c}^T \mathbf{x} = z$$

subject to the constraints:

$$\begin{aligned} x_1a_{11} + \cdots + x_ia_{i1} + \cdots + x_ma_{m1} &\leq b_1 \\ &\vdots \\ x_1a_{1n} + \cdots + x_ia_{in} + \cdots + x_ma_{mn} &\leq b_n \end{aligned}$$

This can be written as:

$$\begin{aligned} \text{maximize } z(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \text{subject to } A\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned} \tag{1}$$

In the two-dimensional case (where  $\mathbf{x} \in \mathbb{R}^2$ ), this is simple since the constraints form a convex polygon; the maximum/minimum value of  $z$  will be found on one of the vertices of the polygon, and in  $\mathbb{R}^3$ , the constraints form a convex polyhedron, where the maximum/minimum value of  $z$  will be found on a vertex.

The objective of this project is to implement in Python one of the most common algorithms to solve the linear programming problem, known as the Simplex Method, and then use the Simplex Method to solve the shortest-path-in-network problem.

### 1.1 The Simplex Method

The Simplex Method changes the inequality  $A\mathbf{x} \leq \mathbf{b}$  into an equality by introducing variables called *slack variables*. These slack variables will contain “leftover” values, so that  $A\mathbf{x} \leq \mathbf{b}$  is still a true inequality. This creates a new matrix equation:

$$(A \ I_m)\hat{\mathbf{x}} = \mathbf{b} \text{ if } A \in \mathcal{M}_{m \times n}, \mathbf{x} \in \mathbb{R}^n, \text{ and } \mathbf{b} \in \mathbb{R}^m$$

where we let  $\hat{\mathbf{x}} = (x_1 \ x_2 \ \dots \ x_n \ s_1 \ s_2 \ \dots \ s_m)^T \in \mathbb{R}^{n+m}$ .  $x_i$  is the  $i^{\text{th}}$  element of the original vector  $\mathbf{x}$  and  $s_j$  is the  $j^{\text{th}}$  slack variable (of which there are  $m$ : one for each row in  $A$ ).

Our system from (1) then becomes:

$$\begin{aligned} z(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ (A \ I_m) \hat{\mathbf{x}} &= \mathbf{b} \\ \hat{\mathbf{x}} &\geq \mathbf{0} \end{aligned} \tag{2}$$

The system in (2) is then put into what is known as a *tableau*:

$$\left( \begin{array}{cc|c} A & I_m & \mathbf{b} \\ -\mathbf{c}^T & \mathbf{0}_m^T & 0 \end{array} \right) \tag{3}$$

where the final row is the row of the objective function. In this system, we choose  $m$  columns on the left side of the augmented matrix to be pivots and use Gaussian elimination to reduce these columns so that they are some permutation of  $I$ . Once this is done, if the objective function row has any negative entries, one pivot column now becomes a non-pivot column and a different column with a negative entry in the objective row is selected to be the pivot column. This process continues until the objective function row has no negative entries. Once this happens, we are done and the tableau is finalized. The optimal solution will be the pivot variables being set to the corresponding value in the right column, and the non-pivot variables being set to 0. In the worst case, the Simplex Method runs in  $\mathcal{O}\left(\binom{n+m}{m}\right)$  time. Our goal will be to understand more in depth how this algorithm works and to implement it in Python.

## 1.2 Shortest-Path-in-Network Problem

One common problem in the context of computer science is the shortest-path-in-graph (a.k.a. network) problem. While algorithms such as Dijkstra's algorithm exist and run much faster asymptotically than the Simplex Method, they do not use linear algebra techniques like the Simplex Method does. Consider a directed graph  $G = (V, E)$  (a set of vertices and edges).  $G$  can be represented as an adjacency matrix  $W$ , where  $w_{i,j}$  is the weight of the edge connecting the  $i^{\text{th}}$  vertex to the  $j^{\text{th}}$  vertex or 0 if there is no edge. If all edges in  $E$  have weight 1 (i.e., the edges are unweighted), the shortest path in the graph is solvable through the Simplex Method. Our goal will be to be able to explain why this is so, and use our Simplex Method implementation to solve this problem for both a small, trivial example as well as a complex example with many vertices and edges.