CS 6200: Algorithmics II	**Fall 2020**
Instructor: Avah Banerjee, S&T.	Scribe: Self
Lecture: 17	**Dates: 10/5**

1 A Review of Geometric Structures

We begin by reviewing some simple geometric structures in the Euclidean, d-dimensions (E^d) space.E^1 is a line, E^2 is a plane, E^3 is a space, and so on. A half-space can be characterized by

$$\sum a_i x_i \leq b \tag{1}$$

A plane in $d = 3$ space is characterized by

$$a_1 x_1 + a_2 x_2 + a_3 x_3 = b \tag{2}$$

While a hyper-plane is the generalization of a plane in d-dimensional space

$$\sum a_i x_i = b \tag{3}$$

You can see from this definition that a line in $d = 2$ space is also a hyper-plane.

Consider the intersection of three planes in d=3 space. The intersection could be either a point or a line. However, in d=4 space, the intersection of three hyper-planes could be a plane, a line, or a point.

We now define the d-simplex as the structure having the minimum number of vertices to create a d-dimensional solid. For example, the 1-simplex is a line, the 2-simplex is a triangle, and the 3-simplex is the tetrahedron. Next we extend the idea of solids and generalize it to d-dimensions through the polytrope. in d=2, polytropes are, of course, flat polygons that we are all familiar with. Finally, we introduce the generalization of an edge as the facet.

2 Linear Programming

Linear programming is a special type of mathematical optimizations which attempt to optimize linear objective functions. The feasible solution set (or feasible region) for a linear objective function lies on a convex polytrope in d-dimensional space. The optimal solution for the linear objective function is the set of vertices on the polytrope which satisfy the inequality constraints. An example of this polytrope is shown for the d=2 space in figure 1.

Linear Programming is popular because once it is established that a problem can be represented as a linear programming one, it is then known that a solution can be found in weak polynomial time $O(poly(m, n, \phi))$ where n is the number of variables, m is the number of constraints, and ϕ is the encoding of the constraint coefficients. The variable ϕ is the reason for the complexity being weak. Algorithms with such performance include the ellipsoid method [1] and the interior-point method [2]. The issue of strong polynomial bounding is still an open problem.

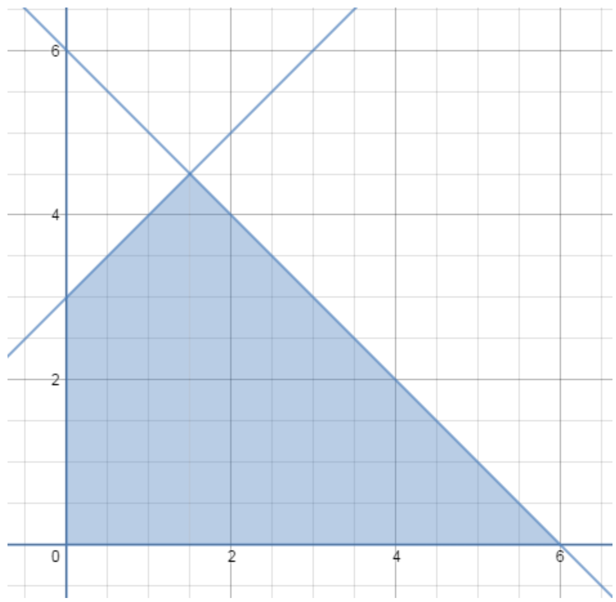

Fig. 1: The Feasible Region

The standard form of a linear programming problem is outlined below.

$$Maximize \quad c^T x \tag{4}$$

Subject to

$$Ax \leq b \tag{5}$$

$$x \geq 0 \tag{6}$$

Where the number of constraints is equal to the size of the vector b. Eq. 4 is the objective function for this problem but not all problems have an objective function. Some books teach the form with with a minimization but both are functionally equivalent.

Common additional constraints are $x \in \mathbb{N}^n$ where the problem becomes integer linear programming and $x \in \{0, 1\}^n$, which makes the problem binary linear programming. Both of these are special cases of linear programming with NP-hard complexity and have their own techniques which will be discussed in future lectures.

2.1 Example: S-T Shortest Path with Linear Programming

Here we provide an example of formulating the S-T shortest path problem for linear programming. We set the objective for our problem to simply be to maximize the distance d_t from the source (S) to the sink (T) subject to the following constraints

$$d_v \leq d_u + w(u,v) \quad \forall (u,v) \in E \tag{7}$$

$$d_s = 0 \tag{8}$$

$$d_u \geq 0 \quad \forall u \in V - \{s\} \tag{9}$$

where d_u and d_v are the distances from the source to the vertices u, v, respectively while d_s is the distance from the source to itself and is, of course, 0. Eq. 8 is not an inequality and therefore must be expressed as one. One such expression is shown in eq. 10.

$$d_s \leq 0, \quad -d_s \leq 0 \tag{10}$$

Every shortest path function must satisfy these constraints. It may seem a contradiction to maximize d_t when we want to find the shortest path but the constraints on the problem actually make it work. Eq. 7 prohibits any paths from S to T from being considered if they are longer than the one already being considered. If the distance function on each of the vertices is given as eq. 11, then we have maximized the variable d_t, where for some $v \in V$, $d_t = d_v$, when d_t is set equal to the minimum of the set.

$$\forall v, \quad d_v = \min_{(u,v) \in E} \{d_u + w(u,v)\} \tag{11}$$

This is but one example of solving a combinatorial problem with linear programming and more examples will come in future lectures.

References and Further Reading

[1] Naum Z Shor. Cut-off method with space extension in convex programming problems.Cybernetics and systems analysis, 13(1):94–96, 1977.

[2] Dantzig, George B.; Thapa, Mukund N. (2003). Linear Programming 2: Theory and Extensions. Springer-Verlag