CS 170 HW 7

Due on 2018-03-12, at 11:59 pm

1 (★) Linear Programming Basics

Plot the feasible region and identify the optimal solution for the following linear program.

$$\text{maximize } 5x + 3y$$
$$5x - 2y \geq 0, \quad x + y \leq 7, \quad x \leq 5, \quad x \geq 0, \quad y \geq 0$$

2 (★★★) Modeling: Tricks of the Trade

One of the most important problems in the field of *statistics* is the *linear regression problem*. Roughly speaking, this problem involves fitting a straight line to statistical data represented by points – $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ – on a graph. Denoting the line by $y = a + bx$, the objective is to choose the constants a and b to provide the "best" fit according to some criterion. The criterion usually used is the *method of least squares*, but there are other interesting criteria where linear programming can be used to solve for the optimal values of a and b. For each of the following criteria, formulate the linear programming model for this problem:

1. Minimize the sum of the absolute deviations of the data from the line; that is,

$$\min \sum_{i=1}^{n} |y_i - (a + bx_i)|$$

   (*Hint:* Define a new variable $z_i = y_i - (a + bx_i)$. Notice that z_i can be either positive or negative. Any number, positive or negative, however, can be represented as the difference of two non-negative numbers. Also define as non-negative variables z_i^+ and z_i^- such that $z_i = z_i^+ - z_i^-$. How can we minimize $|z_i|$ by either minimizing or maximizing some function of z_i^+ and z_i^-?)

2. Minimize the maximum absolute deviation of the data from the line; that is,

$$\min \max_{i=1\ldots n} |y_i - (a + bx_i)|$$

   (*Hint:* You'll need to start by using the same trick as above. Then consider how we can turn our objective function into a single minimization or maximization.)

3 (★★) Spaceship

A spaceship is being designed to take astronauts to Mars and back. This ship will have three compartments, each with its own independent life support system. The key element in each of these life support systems is a small *oxidizer* unit that triggers a chemical process for producing oxygen. However, these units cannot be tested in advance, and only some succeed in triggering this chemical process. Therefore it is important to have several backup units for each system. Because of differing requirements for the three compartments, the units needed for each have somewhat different characteristics. A decision must now be made on just *how many* units to provide for each compartment, taking into account design limitations on the *total* amount of *space*, *weight* and *cost* that can be allocated to these units for the entire ship. The following table summarizes these limitations as well as the characteristics of the individual units for each compartment:

Compartment	Space (cu in.)	Weight (lb)	Cost ($)	Probability of failure
1	40	15	30,000	0.30
2	50	20	35,000	0.40
3	30	10	25,000	0.20
Limitation	500	200	400,000	

The objective is to *minimize the probability* of all units failing in all three compartments, subject to the above limitations and the further restriction that each compartment have a probability of no more than 0.05 that all its units fail.

Formulate the *integer programming model* for this problem. An integer programming model is the same as a linear programming model with the added functionality that variables can be forced to be integers. (*Hint:* Use logarithms.)

Integer programming is often intractable, so we use a linear program as a heuristic. We can take out the integrality constraints and change our model into a linear program. We then round the solution and make sure none of constraints have been violated (you should think about why this won't always give us the optimum)

4 (★★★) Generalized Max Flow

Consider the following generalization of the maximum flow problem.

You are given a directed network $G = (V, E)$ with edge capacities $\{c_e\}$. Instead of a single (s, t) pair, you are given multiple pairs $(s_1, t_1), ..., (s_k, t_k)$, where the s_i are sources of G and t_i are sinks of G. You are also given k demands $d_1, ..., d_k$. The goal is to find k flows $f^{(1)}, ..., f^{(k)}$ with the following properties:

(a) $f^{(i)}$ is a valid flow from s_i to t_i.

(b) For each edge e, the total flow $f_e^{(1)} + f_e^{(2)} + ... + f_e^{(k)}$ does not exceed the capacity c_e.

(c) The size of each flow $f^{(i)}$ is at least the demand d_i.

(d) The size of the *total* flow (the sum of the flows) is as large as possible.

Find a polynomial time algorithm to solve this generalization.

Do not give a four part solution for this problem. Main idea and runtime are sufficient.

5 (★★★★) Reductions Among Flows

Show how to reduce the following variants of Max-Flow to the regular Max-Flow problem, i.e. do the following steps for each variant: Given a graph G and the additional variant constraints, show how to construct a graph G' such that

(1) If F is a flow in G satisfying the additional constraints, there is a flow F' in G' of the same size,

(2) If F' is a flow in G', then there is a flow F in G satisfying the additional constraints with the same size.

Prove that properties (1) and (2) hold for your graph G'.

1. **Max-Flow with Vertex Capacities:** In addition to edge capacities, every vertex $v \in G$ has a capacity c_v, and the flow must satisfy $\forall v : \sum_{u:(u,v)\in E} f_{uv} \leq c_v$.
2. **Max-Flow with Multiple Sources:** There are multiple source nodes $s_1, \ldots, s_k$, and the goal is to maximize the total flow coming out of all of these sources.
3. **Feasibility with Capacity Lower Bounds: (Extra Credit)** In addition to edge capacities, every edge (u, v) has a demand d_{uv}, and the flow along that edge must be at least d_{uv}. Instead of proving (1) and (2), design a graph G' and a number D such that if the maximum flow in G' is at least D, then there exists a flow in G satisfying $\forall(u, v) : d_{uv} \leq f_{uv} \leq c_{uv}$.

6 (★★★★★) A Flowy Metric

Consider an undirected graph G with capacities $c_e \geq 0$ on all edges. G has the property that any cut in G has capacity at least 1. For example, a graph with a capacity of 1 on all edges is connected if and only if all cuts have capacity at least 1. However, c_e can be an arbitrary nonnegative number in general.

1. Show that for any two vertices $s, t \in G$, the max flow from s to t is at least 1.
2. Define the *length* of a flow f to be $\texttt{length}(f) = \sum_{e\in G} |f_e|$. Define the *flow distance* $d_{\texttt{flow}}(s, t)$ to be the minimum length of any $s-t$ flow f that sends one unit of flow from s to t and satisfies all capacities; i.e. $|f_e| \leq c_e$ for all edges e.

   Show that if $c_e = 1$ for all edges e in G, then $d_{\texttt{flow}}(s, t)$ is the length of the shortest path in G from s to t.

   (*Hint*: Let $d(s, t)$ be the length of the shortest path from s to t. A good place to start might be to first try to show $d_{\texttt{flow}}(s, t) \leq d(s, t)$. Then try to show $d_{\texttt{flow}}(s, t) \geq d(s, t)$)
3. **(Extra Credit)** The shortest path satisfies the *triangle inequality*, that is for three vertices, s, t, and u in G, if $d(x, y)$ is the length of the shortest path from x to y, then $d(s, t) \leq d(s, u) + d(u, t)$. Show that the triangle inequality also holds for the flow distance. That is; show that for any three vertices $s, t, u \in G$

$$d_{\texttt{flow}}(s, t) \leq d_{\texttt{flow}}(s, u) + d_{\texttt{flow}}(u, t)$$

   even when the capacities are arbitrary nonnegative numbers.