Problem 1. Linear Programming (50 points)

Let's write the shortest-path problem as an optimization problem. The input is a (directed) graph $G = (V, E)$, weights $w : E \to \mathbb{R}^+$, and two nodes $s, t \in V$. The goal is to find a set of edges $P \subseteq E$ such that P contains a path from s to t and $\sum_{e \in P} w(e)$ is minimized. For simplicity, you may assume that the weights are such that there is exactly one shortest path from s to t. You will now show how to solve this problem using linear programming (even though we know how to solve it using Bellman-Ford or Dijkstra).

(a) Write a *minimization* LP (specifying what the variables are, what the constraints are, and what the objective function is) such that the value of the LP is equal to the total weight of the shortest path. Prove this.

Hint: the book describes a maximization LP for this problem. Your minimization LP should probably be very different, and in particular should probably have a variable for every edge.

Now let's switch from shortest-path to minimum spanning trees. We now have an (undirected) graph $G = (V, E)$ and weights $w : E \to \mathbb{R}^+$. As we discussed in class, one way of phrasing the minimum spanning tree problem is as finding the minimum cost connected subgraph which spans all nodes. This interpretation naturally gives rise to a straightforward LP relaxation which requires every cut to have at least one edge crossing it (fractionally). More formally, suppose that we have a variable x_e for every edge e, and consider the following linear program. For all $S \subset V$, let $\delta(S)$ denote the edges with exactly one endpoint in S and exactly one endpoint not in S.

$$\begin{aligned} \min \sum_{e \in E} w(e) x_e & \\ \text{subject to} \sum_{e \in \delta(S)} x_e \geq 1 & \qquad \forall S \subset V : S \neq \emptyset \\ x_e \geq 0 & \qquad \forall e \in E \end{aligned}$$

Note that there are an exponential number of constraints, but let's not worry about that.

(b) Prove that the optimal value of this LP is at most the weight of the minimum spanning tree.

(c) Unlike the shortest path LP, this is not an exact formulation of the MST problem. Find a graph $G = (V, E)$ and weights $c : E \to \mathbb{R}^+$ such that the optimal LP value is strictly less than the weight of the MST (and prove this).

Part (a)

Solution: We will create a variable x_{uv} for every edge $(u,v) \in E$. Then, we can define the linear program as follows:

$$\begin{aligned} \min \sum_{(u,v)\in E} & w(u,v)x_{uv} & \\ \text{subject to} \sum_{v:(s,v)\in E} & x_{sv} = 1 & \\ \sum_{u:(u,t)\in E} & x_{ut} = 1 & \\ \sum_{u:(u,v)\in E} & x_{uv} - \sum_{u:(v,u)\in E} x_{vu} = 0 & \forall v \in V \setminus \{s,t\} \\ & x_{uv} \geq 0 & \forall (u,v) \in E \end{aligned}$$

Intuitively, this LP corresponds to finding the cheapest way of sending 1 unit of flow from s to t, where the price for sending α flow along edge (u,v) is $\alpha w(u,v)$.

Proof of Correctness: First, observe that any s–t path is a feasible solution. Consider any path $p = s \to v_1 \to v_2 \rightsquigarrow v_n \to t$. We will set $x_{v_i v_{i+1}} = 1$ for all (v_i, v_{i+1}) in p, and $x_{uv} = 0$ otherwise, so this satisfies the nonnegativity constraint. It is easy to see that all other constraints are also satisfied; namely, we have that

$$\begin{aligned} \sum_{v:(s,v)\in E} x_{sv} &= x_{sv_1} = 1, & \\ \sum_{u:(u,t)\in E} x_{ut} &= x_{v_n t} = 1, & \\ \sum_{u:(u,v_i)\in E} x_{uv_i} - \sum_{u:(v_i,u)\in E} x_{v_i u} &= x_{v_{i-1}v_i} - x_{v_i v_{i+1}} = 1 - 1 = 0 & \text{for } v_i \in p \\ \sum_{u:(u,v)\in E} x_{uv} - \sum_{u:(v,u)\in E} x_{vu} &= 0 - 0 = 0 & \text{for } v \notin p. \end{aligned}$$

Thus the optimal LP value is *at most* the total weight of the shortest path. We now need to prove that the optimal value is *at least* the total weight of the shortest path (and thus that they are equal).

It is easy to see that given any $s-t$ flow $\vec{x}$ of value α, we can decompose it into *flow paths*: for each $s-t$ path P, we assign a flow value f_P such that $\sum_P f_P = \alpha$ and $\sum_{P:e\in P} f_P \leq x_e$. This can be proved by induction: given a flow $\vec{x}$, we find an arbitrary $s-t$ path P in which all edges have nonzero flow, and we remove flow from this path until some edge in this path has zero flow (that is, we reduce the flow of all edges in the path by $\beta = \min_{e\in P} x_e$). Clearly what remains is still a valid flow (of value $\alpha - \beta$), so by induction we can decompose this remaining flow into flow paths. We then add P as a flow path, with flow $f_P = \beta$.

So let $\vec{x}$ be an optimal LP solution, and consider the equivalent decomposition into flow paths. The cost of this LP solution is then

$$\sum_{e \in E} w(e)x_e \geq \sum_P f_P \left(\sum_{e \in P} w(e) \right),$$

since $\sum_{P:e \in P} f_P \leq x_e$. Note that $\sum_{e \in P} w(e)$ is exactly the length of P, and thus the right hand side of the above inequality is just taking a weighted average of the lengths of a set of paths (since $\sum_P f_P = 1$). Since any weighted average is at least the minimum value in the average, this implies that the right hand side is at least the length of the shortest path. Hence the value of any LP solution is at least the length of the shortest path, as claimed.

Part (b)

Any feasible solution to the LP is an upper bound on the optimal value, so we just need to prove that the MST is a feasible solution.

Let $T \subseteq E$ be an MST. For every $e \in E$, we have that $x_e = 1$ if $e \in T$ and $x_e = 0$ if $e \notin T$. Thus, the MST satisfies the nonnegativity constraints.

For every cut $(S, V \ S)$, there must be some edge $e \in T$ that crosses $(S, V \ S)$, since otherwise, T would be disconnected, and it wouldn't be a spanning tree. Thus, the MST satisfies the other constraints as well.

Thus, the MST is a feasible solution, and it has objective value of $\sum_{e \in E} w(e)x_e = \sum_{e \in T} w(e)$. Thus the LP optimum is at most the weight of the tree.

Part (c)

As usual, there are many possible answers to this question. Figure 2 shows a simple example. The MST has weight 3 (choose any 3 of the edges), but there is a smaller feasible solution: set $x_e = 1/2$ for all of the edges.

We can prove this by enumeration. There are 14 possible cuts (not including $S = V$ or $S = \emptyset$).

Suppose the cut is $(\{v_1\}, \{v_2, v_3, v_4\})$. Then the edges crossing it are $e_{12} = \{v_1, v_2\}$ and $e_{13} = \{v_1, v_3\}$. We see that $x_{e_{12}} + x_{e_{13}} = 1$.

The same reasoning applies to all the other cuts with $|S| = 1$. It also applies to the cuts where $|S| = 3$. So this covers 8 of the 14 cuts.

Now suppose the cut is $(\{v_1, v_2\}, \{v_3, v_4\})$. Then again, we have two edges crossing the cut, $e_{13} = \{v_1, v_3\}$ and $e_{24} = \{v_2, v_4\}$. Again, we can see that $x_{e_{13}} + x_{e_{24}} = 1$.

The same reasoning applies to all the other cuts with two adjacent nodes. There are four of these. That leaves us with two more cuts – the ones which include edges from opposite corners. Here, all four edges cross the cut, so we will have $x_{e_{12}} + x_{e_{13}} + x_{e_{24}} + x_{e_{34}} = 2 \geq 1$.

Thus, setting $x_e = 1/2$ for all $e \in E$ gives us a feasible solution. This feasible solution has smaller value (2) than the MST (3).

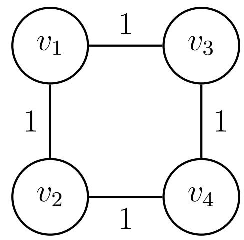

Figure 1: A graph with a feasible solution that is smaller than the MST.

Problem 2. More Cuts and Flows (50 points)

In class we saw the *Multicommodity Flow* problem: given a directed graph $G = (V, E)$, capacities $c : E \to \mathbb{R}^+$, and a collection of k source-sink pairs $\{(s_i, t_i)\}_{i\in[k]}$, maximize the total flow sent (summed over all of the k commodities). In this setting each commodity i must itself be a valid $s_i - t_i$ flow (satisfying flow-balance constraints), while together (summed over all commodities) they must satisfy all edge capacity constraints. We saw how to write a linear program for this problem.

Let's consider a different variant: the *Scaled Multicommodity Flow* problem. We are again given a directed graph $G = (V, E)$, capacities $c : E \to \mathbb{R}^+$, and a collection of k source-sink pairs $\{(s_i, t_i)\}_{i\in[k]}$, but we are also given demands $d : [k] \to \mathbb{R}^+$ for each commodity. Think of $d(i)$ as the amount of commodity i that we *want* to send from s_i to t_i. Of course, it might not be possible to satisfy all of this demand. In that case, we could just send as much as we can, which gets us back the Multicommodity Flow problem. But that might be very unfair to some commodities, who might (for example) get 0 flow while other commodities get their entire demand. To fix this, we will instead try to scale all demands down proportionally until we can actually satisfy this scaled demand.

Slightly more formally, in the Scaled Multicommodity Flow problem our objective is to find the largest value λ such that the it is possible to simultaneously send $\lambda \cdot d(i)$ flow from s_i to t_i for each commodity $i \in [k]$ subject to each commodity obeying the flow-balance constraints, and the total flow (summed over all commodities) satisfying the edge capacity constraints.

(a) Show how to use linear programming to solve the scaled multicommodity flow problem. Be sure to specify what the variables are, what the constraints are, and what the objective function is.

We saw in class that for normal $s - t$ flow, the maximum flow possible is always equal to the capacity of the minimum cut. To study generalizations of this, we need a notion of "cut" which corresponds to the Scaled Multicommodity Flow problem. Recall that in the $s - t$ flow setting, one way to define a cut was a set S of vertices such that $s \in S$ and $t \notin S$, and the capacity of the cut was $\sum_{(u,v)\in E: u\in S, v\notin S} c(u, v)$. To generalize this, we say that a cut is still a partition of the nodes into S and $\bar{S}$, but now the "value" of a cut is the capacity of the cut divided by the demand across the cut. More formally, let

$$val(S) = \frac{\sum_{(u,v)\in E: u\in S, v\notin S} c(u, v)}{\sum_{i\in[k]: s_i\in S, t_i\notin S} d(i)}.$$

If the denominator is 0, we say that the value is infinite.

(b) Prove that on every instance, the maximum scaled multicommodity flow (the largest possible value of λ) is at most $\min_{S \subseteq V} val(S)$.

Unlike the maximum $s-t$ flow and the minimum $s-t$ cut, it turns out that the maximum scaled multicommodity flow might be strictly less than the minimum value cut. Consider the following example, in which all edge capacities are 1: [see assignment for picture].

(c) Prove that the maximum scaled multicommodity flow is 1/2, while the minimum value cut has value 1.

Part (a)

Consider the following LP. There is a variable $x^i_{u,v}$ for each $(u,v) \in E$ and $i \in [k]$. There is also a variable λ. Our LP is

$$\begin{aligned}
\max \quad & \lambda \\
\text{subject to} \quad & \sum_{u:(s_i,u)\in E} x^i_{s_i,u} - \sum_{u:(u,s_i)\in E} x^i_{u,s_i} \geq \lambda d(i) && \forall i \in [k] \\
& \sum_{u:(u,v)\in E} x^i_{u,v} = \sum_{u:(v,u)\in E} x^i_{v,u} && \forall v \in V \setminus \{s_i, t_i\} \\
& \sum_{i\in[k]} x^i_{u,v} \leq c(u,v) && \forall (u,v) \in E \\
& x^i_{u,v} \geq 0 && \forall (u,v) \in E, \forall i \in [k]
\end{aligned}$$

In other words, this is the multicommodity flow LP but instead of maximizing the total flow, we maximize λ subject to all commodities sending at least λ fraction of their demand.

Part (b)

Let $S \subseteq V$ be the set with minimum value, i.e. let S be the set such that $val(S) = \min_{S' \subseteq V} val(S')$. We want to show that the maximum scaled multicommodity flow is at most $val(S)$. Let λ be the value of the maximum scaled multicommodity flow. Then for any $i \in [k]$, this means that if $s_i \in S$ and $t_i \notin S$ then at least $\lambda d(i)$ flow of commodity i crosses S. Thus the *total* flow crossing S is at least

$$\sum_{i\in[k]:s_i\in S, t_i\notin S} \lambda d(i) = \lambda \sum_{i\in[k]:s_i\in S, t_i\notin S} d(i).$$

On the other hand, the total flow across S clearly cannot be more than the total capacity of edges across S. Thus we get that

$$\lambda \sum_{i\in[k]:s_i\in S, t_i\notin S} d(i) \leq \sum_{(u,v)\in E: u\in S, v\notin S} c(u,v),$$

and thus

$$\lambda \leq \frac{\sum_{(u,v)\in E: u\in S, v\notin S} c(u,v)}{\sum_{i\in[k]: s_i\in S, t_i\notin S} d(i)}$$

as claimed.

Part (c)

First consider any cut S. It is easy to see that there is always exactly one edge crossing S. Similarly, there is exactly one unit of demand crossing S: if $|S| = 1$ then this is trivial, but it is also true when $|S| = 2$ (if $S = \{s_1, s_2\}$ then only commodity 2 crosses the cut, if $S = \{s_1, s_3\}$ then only commodity 1 crosses the cut, and if $S = \{s_2, s_3\}$ then only commodity 3 crosses the cut). Thus every cut has value $1/1 = 1$.

Now consider the maximum scaled multicommodity flow. Note that for every commodity, there is only one path on which to send flow, and this path has length exactly 2. Thus we can achieve $\lambda = 1/2$ by sending $1/2$ unit of flow on each path – since every edge is in two paths, capacities are not violated. But if $\lambda > 1/2$, then on each edge the flow is at least $\lambda + \lambda > 1$, which would violate the capacities. Thus the maximum scaled multicommodity flow is $1/2$.