CS 38, PS 7

Codename: Jambul

June 1, 2023

1 Problem 1

Proof. Denote the systems

$$P_1 = \begin{cases} y^T A = c^T \iff A^T y = c, \\ y^T \geq 0 \end{cases} \quad P_2 = \begin{cases} Ax \leq 0, \\ c^T x > 0 \end{cases}.$$

Set $B = A^T$ so we may write the systems as

$$P_1 = \begin{cases} By = c, \\ y^T \geq 0 \end{cases} \quad P_2 = \begin{cases} B^T x \leq 0, \\ c^T x > 0 \end{cases}.$$

We can now see how to apply LP strong duality in order to complete the proof. Consider the primal-dual pair

$$\mathcal{P} = \begin{cases} \min 0y \\ \text{s.t. } By = c, \\ y \geq 0 \end{cases} \quad \mathcal{D} = \begin{cases} \max c^T x \\ \text{s.t. } B^T x \leq 0 \end{cases}.$$

Taking $x = 0$, we can see that $\mathcal{D}$ is trivially feasible. Suppose that $\mathcal{P}$ is infeasible. Then it must be the case that $\mathcal{D}$ is unbounded, since otherwise we would have that $\mathcal{D}$ is feasible and bounded, which would imply from strong duality that $\mathcal{P}$ and $\mathcal{D}$ have equivalent optimal values. This would be a contradiction, since $\mathcal{P}$ is assumed to be infeasible. However, if $\mathcal{D}$ then must be unbounded and feasible, then clearly we can find x such that $c^T x > 0$ and $B^T x \leq 0$, meaning that it cannot be the case that both P_1 and P_2 are infeasible.

Now suppose that $\mathcal{P}$ is feasible. Then there exists $y \in \mathbb{R}^n$ such that $By = c, y \geq 0$. Since $\mathcal{P}$ will then need to be bounded to satisfy its constraints ($By = c$), we have that $\mathcal{P}$ is feasible and bounded, so its optimal value must match that of $\mathcal{D}$. The optimal value of $\mathcal{P}$ will be 0, since $\mathcal{P}$ seeks to minimize $0y$. Thus $\mathcal{D}$ will have to have that its optimal value is $c^T x = 0$. However, this would imply that $c^T x \not> 0$, and thus if P_1 is feasible, then P_2 is not. $\square$

2 Problem 2

Algorithm 1. We will use Ford-Fulkerson algorithm for max flows to solve the problem.

1. Create a new directed graph $G' = (V', E')$ from $G = (V, E)$, so $(u, v) \in E \implies (u, v), (v, u) \in E'$.
2. Set all edge capacities in G' to 1.
3. Pick an arbitrary node $v \in V'$ and compute the maximum flow $g(v, w)$ from v to each $w \in V' \backslash \{v\}$.
4. Return $\min_{w \in V', w \neq v}\{g(v, w)\}$.

Proof of correctness. Let $c = \min_{w \in V', w \neq v}\{g(v, w)\}$. We must show that c is equal to the edge connectivity (let this be k) of G.

Let H be the subset of edges in E such that removal of H, where H has size k, from G will disconnect G into two non-empty subgraphs $G'_1 = (V'_1, E'_1), G'_2 = (V'_2, E'_2)$. Now take any $v \in V'_1, w \in V'_2$. Let $g(v, w)$ then be the calculated max flow from v to w with all edge capacities in E' set to 1.

From the max-flow min-cut theorem, then, we know that $g(v, w)$ will also be the size of a min cut between vertices v and w. Note furthermore that the size of a min cut between these two vertices is at most k, since if it takes at minimum k edges to get G'_1, G'_2 from G, then a min cut (and thus a max flow) between v and w should be of size less than or equal to k. From this observation, we now have that $c \leq g(v, w) \leq k$.

However, it must also be the case that $c \geq k$, since otherwise there exists a cut of G with size smaller than k, which would be a contradiction. Thus we have that in fact $c = g(v, w) = k$, as desired.

Complexity. G' has $|V|$ vertices and $2|E|$ edges. It then solves a max-flow problem from v to every other $w \in V' \backslash \{v\}$, so it solves $|V| - 1$ max-flow problems on a network with $\mathcal{O}(|V|)$ vertices and $\mathcal{O}(|E|)$ edges.