**Problem 1:** (10 points) In a mixed-strategy matrix game, the best strategy for the player who goes second is a pure strategy. Illustrate this using the matrix $M$ below, with the first player being the row player using the mixed strategy $\vec{p}$ below.

$$M = \begin{bmatrix} 10 & -30 & 10 \\ 0 & 10 & -10 \end{bmatrix} \quad \vec{p} = \begin{bmatrix} .4 \\ .6 \end{bmatrix}$$

**Solution:** Denote the mixed strategy of the second player—the column player—as $\vec{q} = [q_1, q_2, q_3]^T$. The expected value of the game is

$$q_1 \left( .4 \cdot 10 \right) + q_2 \left( .4 \cdot -30 \right) + q_3 \left( .4 \cdot 10 \right) = q_1(4) + q_2(-6) + q_3(-2)$$

Which is a convex combination of the three values $4, -6, -2$, and the best choice for the second player for minimization is to most heavily weight the minimum of these values $-6$, which in this case is to take $q_2$ to be 1, which is a pure strategy. (In general, $q_i = 1$ for $i \in \arg\min_j (\vec{p}^T M)_j$ is the best strategy, and it is a pure strategy.)

**Problem 2:** (10 points) Write the advantages of Dijskstra’s Algorithm over the Label Correcting Algorithm, and vice versa.

**Solution:** Dijkstra’s algorithm runs in nearly linear time (with proper data structures), whereas the Label Correcting Algorithm (scan implementation) runs in time on the order of the number of nodes times number of arcs. In Dijkstra’s Algorithm, even in the middle of the algorithm, Found nodes are optimally solved, whereas the Label Correcting Algorithm needs to terminate before we have confidence in the optimality for any of the nodes. On the other hand, Dijsktra’s Algorithm requires that all arcs have nonnegative length, whereas the Label Correcting Algorithm allows for negative length arcs, provided that there are no negative length cycles.
Problem 3: (10 points) Let $M$ be any particular matrix. State what you know (proof not necessary) about the relationship between the value of the max min version and the value of the min max version of the matrix game with $M$, for the matrix game scenarios and cases we studied.

Solution: It always holds that the value of the max min version is less than or equal to the value of the min max version. In the case of pure strategy, equality in value between the max min and the min max versions holds if and only if $M$ has a saddle point, and in the case of mixed strategy we always have equality in value between the max min and the min max versions.

Problem 4: (10 points) Suppose the Ford-Fulkerson Algorithm is applied to max flow min cut instance $G = (V, E), s, t \in V, \mu : E \to \mathbb{R}_{\geq 0}$. When it terminates, the algorithm outputs a feasible flow assignment $x$ such that there is no $s, t$ path in $G^x$, and it outputs $B$, which are the reachable nodes in $G^x$ from $s$. Prove that $x$ is a max flow and that $[B, B^C]$ is a min cut.

Solution: Note that $s \in B$ and $t \notin B$. Now, for all $(u, v) \in [B, B^C]$ we have $\mu^x(e) = 0$ (if not, then $(u, v) \in G^x$, and then $u$ reachable from $s$ in $G^x$ implies $v$ reachable from $s$ in $G^x$, i.e. $v \in B$, a contradiction). Next, summing residual capacities over all arcs in $[B, B^c]$, we get

$$0 = \sum_{e \in [B, B^C]} \mu^x(e) = \sum_{e \in [B, B^C]} (\mu(e) - x(e) + x(e^{-1}))$$

$$= \sum_{e \in [B, B^C]} \mu(e) - \sum_{e \in [B, B^C]} (x(e) - x(e^{-1})) = \mu[B, B^C] - \text{effective } s, t \text{ flow value of } x,$$

Thus the effective $s, t$ flow value of $x$ equals the capacity of the $s, t$ cut $[B, B^C]$, hence each of $x$ and $[B, B^C]$ are optimal in their respective problems by the supervisor principle.
Problem 5: (10 points) Consider the MaxFlow-MinCut problem instance—and a feasible flow $x$—illustrated below; each arc $e$ is labeled with $x(e) / \mu(e)$. (Note: “/” is a separator, not a fraction.)

a) Write out the residual network $G^x$.

b) Either demonstrate that there is no augmenting path and this flow $x$ is optimal, or clearly identify an augmenting path and update the flow $x$ to a new flow.
SCRAP PAPER This page will not be graded.