CS 170 Homework 10

Due **Saturday 4/12/2025, at 10:00 pm (grace period until 11:59pm)**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, explicitly write "none".

2 Applications of Max-Flow Min-Cut

Review the statement of max-flow min-cut theorem and prove the following two statements.

(a) Let $G = (L \cup R, E)$ be a unweighted bipartite graph[1]. Then G has a L-perfect matching (a matching[2] with size $|L|$) if and only if, for every set $S \subseteq L$, S is connected to at least $|S|$ vertices in R. You must prove both directions.

Hint: Use the max-flow min-cut theorem on the cut that forms S and $L \setminus S$.

(b) Let G be an unweighted directed graph and $s, t \in V$ be two distinct vertices. Then the maximum number of edge-disjoint s-t paths equals the minimum number of edges whose removal disconnects t from s (*i.e.*, no directed path from s to t after the removal).

Hint: show how to decompose a flow of value k into k disjoint paths, and how to transform any set of k edge-disjoint paths into a flow of value k.

Solution:

(a) The proposition is known as Hall's theorem. On one direction, assume G has a perfect matching, and consider a subset $S \subseteq L$. Every vertex in S is matched to distinct vertices in R, so in particular the neighborhood of S is of size at least $|S|$, since it contains the vertices matched to vertices in S.

On the other direction, assume that every subset $S \subseteq L$ is connected to at least $|S|$ vertices in R. Add two vertices s and t, and connect s to every vertex in L, and t to every vertex in R. Let each edge have capacity one. We will lower bound the size of any cut separating s and t. Let C be any cut, and let $L = S \cup Y$, where S is on the same side of the cut as s, and Y is on the other side. There is an edge from s to each vertex in Y, contributing at least $|Y|$ to the value of the cut. Now there are at least $|S|$ vertices in R that are connected to vertices in S. Each of these vertices is also connected to t, so regardless of which side of the cut they fall on, each vertex contributes one edge cut (either the edge to t, or the edge to a vertex in S, which is on the same side as s). Thus the cut has value at least $|S| + |Y| = |L|$, and by the max-flow min-cut theorem, this implies that the max-flow has value at least $|L|$, which implies that there must be a perfect matching.

[1] A bipartite graph $G = (L \cup R, E)$ is defined as a graph that can be partitioned into two disjoint sets of vertices (i.e. L and R) such that no two vertices within the same set are adjacent.

[2] A matching is defined as a set of edges that share no common vertices.

(b) The proposition is known as Menger's theorem. By max-flow min-cut theorem, we only need to show that the max flow value from s to t equals the maximum number of edge-disjoint s-t paths.

If we give each edge capacity 1, then the maxflow from s to t assigns a flow of either 0 or 1 to every edge (using, say, Ford-Fulkerson). Let F be the set of saturated edges; each has flow value of 1. Then extracting the edge-disjoint s-t paths from the flow can be done algorithmically. Follow any directed path in F from s to t (via DFS), remove that path from F, and recurse. Each iteration, we decrease the flow value by exactly 1 and find 1 edge-disjoint s-t path.

Conversely, we can transform any collection of k edge-disjoint paths into a flow by pushing one unit of flow along each path from s to t; the value of the resulting flow is exactly k.

3 Max-Flow Duality

In lecture, we discussed how the max-flow problem can be described as an LP and also discussed its relation to the min-cut problem. In this problem, we will connect the max-flow and min-cut problems through LPs directly. For the duration of this problem, we will consider a weighted directed graph $G = (V, E)$ with $s, t \in V$ and capacity weights $w : E \to \mathbb{Z}^{>0}$, and we use the following LP formulation of max-flow (that is in canonical form):

$$
\begin{aligned}
\max \quad & \sum_{\substack{v \in V \\ (s,v) \in E}} f_{(s,v)} \\
\text{such that:} \quad & \sum_{\substack{u' \in V \\ (v_0,u') \in E}} f_{(v_0,u')} - \sum_{\substack{u \in V \\ (u,v_0) \in E}} f_{(v_0,u)} \le 0 \qquad \forall v_0 \in V \setminus \{s,t\} \\
& \sum_{\substack{u \in V \\ (u,v_0) \in E}} f_{(v_0,u)} - \sum_{\substack{u' \in V \\ (v_0,u') \in E}} f_{(v_0,u')} \le 0 \qquad \forall v_0 \in V \setminus \{s,t\} \\
& f_{(u,v)} \le w_{(u,v)} \qquad \forall (u,v) \in E \\
& f_{(u,v)} \ge 0 \qquad \forall (u,v) \in E
\end{aligned}
$$

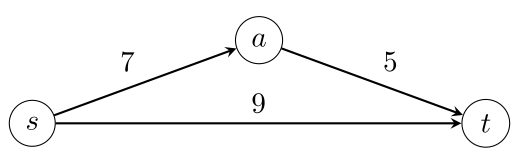
(a) Briefly explain the optimization function and each constraint.

(b) Take the dual of this LP. *Hint: you will have many equation types. This is expected!*

(c) Call the dual LP L. Note that each vertex has potentially multiple variables corresponding to them. Explain how we can group some of the variables to change the dual into a non-canonical form LP, L', that involves one of the variable types from HW8 Q4. *Hint: L' should be slightly cleaner.*

(d) Suppose that (for this part only) we enforced the constraint that each of our variables in L' are either 0 or 1. Give an interpretation for each of the variables of L'. *Hint: start with the vertex variables. Make sure to be careful with + and −!*

(e) Explain why every optimal solution to the L' LP for the following graph assigns edge variables to either 0 or 1. *Hint: you may have to consider the edge of weight 9 separately.*



(f) Prove that, for every arbitrary directed graph (with assigned nodes s and t), there exists an optimal solution to its L' LP assigning all edge variables to either 0 or 1. *Hint: work with edges close to s first.*

(g) Explain how the dual and its optimal solution relate to the min-cut problem.

Solution:

(a) Each $f_{(u,v)}$ corresponds to the amount of flow that we push through each edge. The optimization function is the amount of flow out of s, which is the same as the amount of flow from s to t (since there are no other sources or sinks). The first two equations ensure that the flow into every vertex besides s and t is equal to the flow out of the vertex (i.e. no other vertex is a source or sink). The $f_{(u,v)} \leq w_{(u,v)}$ constraints ensure that the amount of flow through an edges never exceeds capacity. Lastly, the $f_{(u,v)}$ constraints ensure that the flow through an edge is never negative (as that makes no sense!).

(b) Let z variables correspond to flow conservation constraints and y variables correspond to capacity constraints:

$$\begin{aligned}
\min \sum_{(u,v)\in E} & w_{(u,v)} y_{(u,v)} \\
\text{such that: } & y_{(s,v)} + z_{\to v} - z_{v\to} \geq 1 \qquad \forall (s,v) \in E, v \neq t \\
& y_{(u,t)} - z_{\to u} + z_{u\to} \geq 0 \qquad \forall (u,t) \in E, u \neq s \\
& y_{(s,t)} \geq 1 \qquad \text{if } (s,t) \in E \\
& y_{(u,v)} - z_{\to u} + z_{u\to} + z_{\to v} - z_{v\to} \geq 0 \qquad \forall (u,v) \in E, u \neq s, v \neq t \\
& y_{(u,v)} \geq 0 \qquad \forall (u,v) \in E \\
& z_{\to v}, z_{v\to} \geq 0 \qquad \forall v \in V \setminus \{s,t\}
\end{aligned}$$

(c) We'll group $p_v = z_{\to v} - z_{v\to}$, which will be an unrestrained variable:

$$\begin{aligned}
\min \sum_{(u,v)\in E} & w_{(u,v)} y_{(u,v)} \\
\text{such that: } & y_{(s,v)} + p_v \geq 1 \qquad \forall (s,v) \in E, v \neq t \\
& y_{(u,t)} - p_u \geq 0 \qquad \forall (u,t) \in E, u \neq s \\
& y_{(s,t)} \geq 1 \qquad \text{if } (s,t) \in E \\
& y_{(u,v)} + p_v - p_u \geq 0 \qquad \forall (u,v) \in E, u \neq s, v \neq t \\
& y_{(u,v)} \geq 0 \qquad \forall (u,v) \in E
\end{aligned}$$

(d) The vertex variables p_v being 1 or 0 naturally partitions V. To interpret edge variable $y_{(u,v)}$, fix a $(u,v) \in E$. If $p_u = 1$ and $p_v = 0$, then $y_{(u,v)} = 1$. That is, $y_{(u,v)}$ records whether an edge crosses the partition of V. We also have that $y_{(u,t)} = 1$ if $p_u = 1$, meaning that the optimal duals will have $p_u = 1$ when $(u,t) \in E$. We can therefore interpret $\{v \in V \mid p_v = 1\}$ as the vertices on the t side of the cut and $y_{(u,v)}$ as recording whether the capacity for a cut edge is "paid". We can make this explicit by fixing $p_s = 0$ and $p_t = 1$ without loss of generality.

(e) Let's first fix the vertex variables to be 0 or 1. We then observe that, for any realization of vertex variables, all constraints can be satisfied with some choice of $y_{(u,v)} \in \{0,1\}$. Moreover, for any choice of $p \in \{0,1\}$, if $y_{(u,v)} \in (0,1)$ satisfies any given constraint, $y_{(u,v)} = 0$ also satisfies said constraint. Verifying that all constraints are increasing in y concludes the claim.

To verify this on the graph, we can just solve for the optimum explicitly. We want to minimize the dual objective $7y_{(s,a)} + 9y_{s,t} + 5y_{a,t}$. Our constraints are that $p_s = 0$,

$p_t = 1$, $y_{(s,t)} \geq 1$, $y_{s,a} = 0$, and $y_{(a,t)} \geq p_a$. At the optimum, $p_a = 1$, giving that $y_{a,t} = 1$.

(f) Fix an optimal dual solution p^*, y^*. Let's first lower bound y: $y^*_{s,v} = \max\{0, 1 - p_v\}$, $y^*_{u,t} = \max\{0, p_u\}$, $y^*_{s,t} = 1$, $y^*_{u,v} = \max\{0, p_u - p_v\}$. Substituting this into the objective, we have

$$\begin{aligned}\min_{p^*} \sum_{(s,v)\in E, v\neq t} & w_{s,v} \max\{0, 1 - p^*_v\} + \sum_{(u,t)\in E, u\neq s} w_{u,t} \max\{0, p^*_u\} \\ &+ \sum_{(u,v)\in E, u\neq s, v\neq t} w_{u,v} \max\{0, p^*_u - p^*_v\} + w_{s,t}\end{aligned}$$

Note that we can without loss of generality truncate all $p^* \in [0, 1]$. Since this objective is the sum of piecewise linear convex functions and the subgradients of these terms can only be zero in $[0, 1]$ at $\{0, 1\}$, it follows that $p^* \in \{0, 1\}$.

(g) The dual objective is just the cost of a cut. The dual constraints (as mentioned in part d) enforce that vertex variables p_v indicate cut membership and the edge variables $y_{(u,v)}$ indicate edges crossing from the s-side into the t-side. Duality tells us min-cut always upper bounds max-flow; strong duality tells us these quantities are equivalent.

4 Domination

In this problem, we explore a concept called *dominated strategies*. Consider a zero-sum game with the following payoff matrix for the row player:

Column:

		A	B	C
Row:	D	1	2	-3
	E	3	2	-2
	F	-1	-2	2

Note: All parts of this problem can and should be solved without using an LP solver or solving a system of linear equations.

(a) If the row player plays optimally, can you find the probability that they pick D without directly solving for the optimal strategy? Justify your answer.

(b) Given the answer to part a, if the both players play optimally, what is the probability that the column player picks A? Justify your answer.

(c) Given the answers to part a and b, what are both players' optimal strategies?

Solution:

(a) 0. Regardless of what option the column player chooses, the row player always gets a higher payoff picking E than D, so any strategy that involves a non-zero probability of picking D can be improved by instead picking E.

(b) 0. We know that the row player is never going to pick D, i.e. will always pick either E or F. But in this case, picking B is always better for the column player than picking A (A is only better if the row player picks D). That is, conditioned on the row player playing optimally, B dominates A.

(c) Based on the previous two parts, we only have to consider the probabilities the row player picks E or F and the column player picks B or C. Looking at the 2-by-2 submatrix corresponding to these options, it follows that the optimal strategy for the row player is to pick E and F with probability 1/2, and similarly the column player should pick B, C with probability 1/2.

5 Weighted Rock-Paper-Scissors

For this problem only, you are allowed to use code or online tools to automatically solve your LPs. Feel free to use an LP solver such as: `https://online-optimizer.appspot.com/`. However, **make sure to cite your LP solver** if you use one.

You and your friend used to play rock-paper-scissors, and have the loser pay the winner 1 dollar. However, you then learned in CS170 that the best strategy is to pick each move uniformly at random, which took all the fun out of the game.

Your friend, trying to make the game interesting again, suggests playing the following variant: If you win by beating rock with paper, you get 2 dollars from your opponent. If you win by beating scissors with rock, you get 1 dollars. If you win by beating paper with scissors, you get 4 dollar.

(a) Draw the payoff matrix for this game. Assume that you are the maximizer, and your friend is the minimizer.

(b) Write an LP to find the optimal strategy in your perspective. You do not need to solve the LP.

The following subparts are independent of the previous subparts.

Your friend now wants to make the game even more interesting and suggests that you assign points based on the following payoff matrix:

		Your friend: rock	paper	scissors
You:	rock	-10	3	3
	paper	4	-1	-3
	scissors	6	-9	2

(c) Write an LP to find the optimal strategy for yourself. What is the optimal strategy and expected payoff?

Feel free to use an online LP solver.

(d) Now do the same for your friend. What is the optimal strategy and expected payoff? How does the expected payoff compare to the answer you get in part (c)?

Solution:

(a)

		Your Friend: rock	paper	scissors
You:	rock	0	-2	1
	paper	2	0	-4
	scissors	-1	4	0

(b) Let r, p, s be the probabilities that you play rock, paper, scissors respectively. Let z stand for the expected payoff, if your opponent plays optimally as well.

$$
\begin{aligned}
\max \quad & z \\
2p - s &\geq z && \text{(Opponent chooses rock)} \\
4s - 2r &\geq z && \text{(Opponent chooses paper)} \\
r - 4p &\geq z && \text{(Opponent chooses scissors)} \\
r + p + s &= 1 \\
r, p, s &\geq 0
\end{aligned}
$$

(c)

$$
\begin{aligned}
\max \quad & z \\
-10r + 4p + 6s &\geq z && \text{(Opponent chooses rock)} \\
3r - p - 9s &\geq z && \text{(Opponent chooses paper)} \\
3r - 3p + 2s &\geq z && \text{(Opponent chooses scissors)} \\
r + p + s &= 1 \\
r, p, s &\geq 0
\end{aligned}
$$

The optimal strategy is $r = 0.3346,\ p = 0.5630,\ s = 0.1024$ for an optimal payoff of -0.48.

See the coding solution for how to express and solve this LP in Python.

(d)

$$
\begin{aligned}
\min \quad & z \\
-10r + 3p + 3s &\leq z && \text{(You choose rock)} \\
4r - p - 3s &\leq z && \text{(You choose paper)} \\
6r - 9p + 2s &\leq z && \text{(You choose scissors)} \\
r + p + s &= 1 \\
r, p, s &\geq 0
\end{aligned}
$$

Your friend's optimal strategy is $r = 0.2677,\ p = 0.3228,\ s = 0.4094$. The value for this is -0.48, which is the payoff for you. The payoff for your friend is the negative of your payoff, i.e. 0.48, since the game is zero-sum.

See the coding solution for how to express and solve this LP in Python.

(Note for grading: Equivalent LPs are of course fine. It is fine for part (d) to maximize your friend's payoff instead of minimizing yours. For the strategies, fractions or decimals close to the solutions are fine, as long as the LP is correct.)