CS 670 (Spring 2025) — Assignment 4

Due: 04/09/2022

(1) [10 points]
You are given a directed graph G with non-negative edge costs c_e, and a root node $r \in V$, as well as a set $X \subseteq V$ of target nodes. The goal is to find a set F of edges of minimum total cost, such that F contains a directed path from r to each node $v \in X$. (Let's say that at least one such path always exists.)

Phrase this problem as a decision problem and prove that it is NP-complete.

(2) [10 points]
Problem 8.39 from the textbook.

(Hint: I would recommend 3SAT. The reduction is not super easy — the most similar you have seen so far in class would be 3DM, though reading the textbook section on HAMILTONIAN PATH/CYCLE could also be helpful.)

(3) [4+5+3=12 points]
We said in class that Integer Linear Programming (ILP) is very powerful, and can easily encode lots of NP-hard problems. Here, you will explore this for VERTEX COVER.

(a) Write the MINIMUM VERTEX COVER problem as an ILP. That is, for any given MINIMUM VERTEX COVER input (graph G), produce an ILP whose optimal solution encodes a minimum vertex cover. Notice that implicitly, your reduction thus shows that ILP is NP-hard, but you don't need to explicitly phrase it as a Karp Reduction.

(b) Now get rid of the integrality constraint to obtain the fractional version of MINIMUM VERTEX COVER. Compute the dual linear program of this fractional LP. Then, add an integrality constraint to the dual LP, and come up with an interpretation of the resulting dual ILP. That is, what problem does the ILP encode? (Notice that this is similar to what we did for the dual of MAXIMUM FLOW, where the dual with an integrality constraint naturally encoded MINIMUM CUT.)

(c) Combining the "obvious" inequalities between integer and fractional LPs with weak duality, what inequality involving vertex cover sizes have you derived? Now that you have seen this inequality, find a simple "direct" proof (without LP duality).

(4) [5+10=15 points]
In class, we saw the fractional linear program for MINIMUM s-t CUT. Here it is again:

$$\begin{array}{llcll} \text{Minimize} & \sum_{e \in E} x_e \cdot c_e & & & \\ \text{subject to} & \sum_{e \in P} x_e & \geq & 1 & \text{for all } s\text{-}t \text{ paths } P \\ & x_e & \geq & 0 & \text{for all edges } e. \end{array}$$

In the integer program, all the x_e are restricted to be 0 or 1. In the fractional version, they may take values in between. (In an optimum solution, they will never be larger than 1.) Here, you will explore some interesting properties of this Linear Program.

(a) First, observe that the LP has exponentially many constraints. In class, we mentioned that some LPs with exponentially many constraints can nonetheless be solved in polynomial time, if you can provide polynomial-time Membership and Separation Oracles. Recall that a *Membership Oracle* is an algorithm which, given a proposed solution $\boldsymbol{x} = (x_e)_{e \in E}$, will always correctly tell you whether $\boldsymbol{x}$ satisfies all constraints of the LP or not. A *Separation Oracle* will give you one violated constraint if there exists one.

For the specific LP above (the Minimum s-t cut LP), give polynomial-time Membership and Separation Oracles, and brief correctness proofs and runtime analysis.

(b) We also mentioned in class that this LP happens to always have integer optimal solutions; that is, while there may be optimal non-integer solutions as well, there will always be at least one optimal solution in which all x_e are integers. While this follows using the more general "hammer" of total unimodularity, you will prove it directly in this problem.

Prove that the Minimum-Cut LP always has an integer optimal solution.

(Hint: Interpret the x_e values as edge lengths. Then, for each node v, you can assign (and compute) a distance $d(s, v)$ from s. Prove that there must be a value r (which you can find in polynomial time) such that the cut separating $\{v \mid d(s, v) \leq r\}$ from $\{v \mid d(s, v) > r\}$ is an s-t cut and has cost no more than the fractional LP-solution you started with.)

(5) [0 points]
Chocolate Problem (1 chocolate bar): Let's return to the toll road question from Homework 2. At the time, all drivers were going to the *same* final destination. That is of course very unrealistic. Here, we consider the more realistic version where different drivers have different destinations. Here, is a formal description again.

You have m ramps to get on/off your toll road, numbered $1, 2, \ldots, m$. This divides your toll road into segments $T_j = [j, j + 1)$. You are given a list of n drivers. For each driver i, you are told the starting and ending ramp of their trip $s_i < t_i$, as well as a budget b_i. To drive from s_i to t_i, the driver has to drive along all the intermediate segments $T_{s_i}, T_{s_i+1}, \ldots, T_{t_i-1}$.

You get to choose a price for each segment T_j. If the sum of prices on the trip is affordable to the driver (at most b_i), then they will drive on your toll road and pay you the total for the segments. If the sum exceeds b_i, driver i will take surface streets for the entire trip, and not pay you anything. Your goal is to choose prices for the segments that maximize your revenue from the drivers under this model.

Phrase this problem as a decision problem, and prove that it is NP-complete.