1 Multicut in Trees (33 points)

Consider the multicut problem in trees. In this problem, we are given a tree $T = (V,E)$, $k$ pairs $(s_i, t_i)$ of vertices, and edge costs $c : E \to \mathbb{R}^+$. A feasible solution is a set $F \subseteq E$ such that for all $i \in [k]$, $s_i$ and $t_i$ are in different connected components of $T - F$. The objective is to minimize the total edge cost $\sum_{e \in F} c(e)$.

Let $P_i$ be the unique path between $s_i$ and $t_i$ in $T$. Then we can write an integer linear programming formulation of this problem:

$$\begin{align*}
\min & \sum_{e \in E} c(e)x_e \\
\text{subject to} & \sum_{e \in P_i} x_e \geq 1 & \forall i \in [k] \\
& x_e \in \{0, 1\} & \forall e \in E
\end{align*}$$

(a) Write the dual of the LP relaxation of the above ILP

Suppose that we root the tree at an arbitrary vertex $r$. Let $depth(v)$ be the number of edges on the path from $v$ to $r$. Let $lca(s_i, t_i)$ be the vertex $v$ on the path from $s_i$ to $t_i$ whose depth is minimum. Suppose that we use the primal-dual method to solve this problem, where the dual variable that we increase in each iteration corresponds to the violated constraint that maximized $depth(lca(s_i, t_i))$.

(b) Prove that this is a 2-approximation.

2 Shortest Path (33 points)

Prove that the primal-dual shortest $s - t$ path algorithm from class (and from Section 7.3 of the book) is equivalent to Dijkstra’s algorithm. In other words, prove that at each step, the primal-dual algorithm and Dijkstra’s algorithm add the same edge.

3 Maximum Directed Cut (34 points)

Recall the maximum directed cut problem from HW3: given a directed graph $G = (V,E)$ and a nonnegative weight $w_{ij} \geq 0$ for each edge $(i, j) \in E$, the goal is to partition $V$ into two sets $U$ and $W = V \setminus U$ in order to maximize the total weight of edges $(i, j)$ with $i \in U$ and $j \in W$ (that is, the total weight of edges from $U$ to $W$). In HW3 we used randomized rounding of an LP relaxation to give a 1/2-approximation. Now we will improve on that using an SDP relaxation.
(a) Write a strict quadratic program for MAX DICUT (a quadratic program using only quadratic
and constant terms). Relax this to a vector program to write an SDP relaxation for MAX
DICUT. Hint: think about the 2SAT SDP from class. In particular, think about having a
±1 variable for each vertex and an extra variable $y_0$ (in the quadratic program) or $v_0$ (in the
SDP relaxation) to indicate the side corresponding to $U$.

We will now use random hyperplane rounding: we uniformly at random draw a unit vector $r$,
and let $U = \{i : \text{sign}(v_i \cdot r) = \text{sign}(v_0 \cdot r)\}$. Let $\theta_{ij}$ denote the angle between $v_i$ and $v_j$

(b) Prove that the expected weight from $U$ to $W$ is $\sum_{(i,j) \in E} w_{ij} \frac{1}{2\pi} (\theta_{0i} + \theta_{0j} + \theta_{ij})$.

(c) Prove that random hyperplane rounding is a 0.796-approximation.

Hint: the following geometric fact might be helpful. Let $v_i, v_j, v_k$ be unit vectors in $n$ dimen-
sions. Then

$$2 \pi \left( \frac{- \arccos(v_k \cdot v_i) + \arccos(v_k \cdot v_j) + \arccos(v_i \cdot v_j)}{1 + (v_k \cdot v_i) - (v_k \cdot v_j) - (v_j \cdot v_j)} \right) > 0.796$$