HW 10

Due: December 4th, 2025

Problem 1:

In this problem, we consider an unweighted version of the minimum multicut problem, generalizing the problem from Homework 8 from trees to general graphs. We are given a graph $G = (V, E)$ and k pairs of source-sink vertices, $s_i, t_i \in V$ for $i = 1, \ldots, k$. We wish to find a subset of edges $F \subseteq E$ that minimizes $|F|$ such that for each $i = 1, \ldots, k$, there is no s_i-t_i path in $(V, E - F)$.

Consider the following vector program:

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in E} (1 - \vec{v}_i \cdot \vec{v}_j) & \\ \text{subject to} \quad & \vec{v}_{s_i} \cdot \vec{v}_{t_i} = 0, & i = 1, \ldots, k, \\ & \vec{v}_j \cdot \vec{v}_j = 1, & \forall j \in V, \\ & \vec{v}_j \in \mathbb{R}^n, & \forall j \in V. \end{aligned}$$

Consider the *demand graph* $H = (V, E')$, where $E' = \{(s_i, t_i) : i = 1, \ldots, k\}$. Let Δ be the maximum degree of a vertex in the demand graph. Suppose that the optimal value of the vector programming relaxation is $\epsilon|E|$. Consider the following algorithm where we draw $t = \lceil \log_2(\Delta/\epsilon) \rceil$ random vectors $\vec{r}_1, \ldots, \vec{r}_t$. The t random vectors define 2^t different regions into which the vectors $\vec{v}_j$ can fall: one region for each distinct possibility of whether $\vec{r}_\ell \cdot \vec{v}_j \geq 0$ or $\vec{r}_\ell \cdot \vec{v}_j < 0$ for all $\ell = 1, \ldots, t$. Remove all edges (i, j) from the graph such that $\vec{v}_i$ and $\vec{v}_j$ are in different regions. If for any s_i-t_i pair, there still exists an s_i-t_i path, remove all edges incident on s_i. We now analyze this algorithm.

(a) Prove that the vector program is a relaxation of the unweighted minimum multicut problem.

(b) For any $(i, j) \in E$, prove that the probability that i and j are in different regions is at most $t \cdot \sqrt{1 - \vec{v}_i \cdot \vec{v}_j}$.

(c) Prove that for any $i = 1, \ldots, k$, the probability that we end up removing all the edges incident on s_i (because a path remains) is at most $\Delta \cdot 2^{-t}$.

(d) Show that the expected number of edges removed is at most $O(\sqrt{\epsilon} \log(\Delta/\epsilon)|E|)$.

For the final item, it may be useful to use *Jensen's inequality*, which states that for any convex function f (that is, $f''(z) \geq 0$) and any positive p_i,

$$f\left(\frac{\sum_i p_i x_i}{\sum_i p_i}\right) \leq \frac{1}{\sum_i p_i} \sum_i p_i f(x_i).$$

Problem 2: Problem: Worst-Case vs. Smoothed Analysis of Greedy Scheduling

Consider the following scheduling problem, which is the same as Homework 1: We are given n jobs to be scheduled on a single machine. Each job $j \in \{1, \ldots, n\}$ has a processing time $p_j > 0$, a weight $w_j > 0$, and a due date $d_j > 0$. The objective is to find a non-preemptive schedule that maximizes the total weight of the jobs that finish on or before their due date.

We will analyze the following **Greedy-Density Algorithm ($\mathcal{A}$)**:

1. Sort the jobs in decreasing order of their "density," $\rho_j = w_j/p_j$.
2. Initialize the machine's completion time $C = 0$ and a set of scheduled jobs $S = \emptyset$.
3. Iterate through the jobs j in their sorted order:
   - If $C + p_j \leq d_j$, schedule the job by setting $C \leftarrow C + p_j$ and $S \leftarrow S \cup \{j\}$.
4. Return the total weight of the scheduled jobs, $W_{\mathcal{A}} = \sum_{j \in S} w_j$.

Part A. Worst-Case Analysis

Prove that Algorithm $\mathcal{A}$ is **not** a constant-factor approximation algorithm. To do this, construct a family of problem instances (e.g., using two jobs) and show that for any $c > 1$, you can choose parameters such that the ratio $W_{OPT}/W_{\mathcal{A}} > c$.

Part B. Analysis of a Perturbed Instance

The hard instances from Part A rely on a "brittle" alignment of due dates. Let's analyze this using a smoothed analysis model.

First, define the perturbation model: An adversary provides an arbitrary instance $I = \{(p_j, w_j, d_j)\}_{j=1}^n$. A *perturbed instance* I' is created by replacing each d_j with a new random due date d'_j, drawn independently from a uniform

distribution over the interval $[d_j, d_j + \sigma P]$, where $P = \sum_{k=1}^{n} p_k$ and $\sigma > 0$ is a perturbation parameter.

Take the instance I you constructed in Part A. Let I' be its perturbed version under this model.

1. Calculate the expected total weight $\mathbb{E}[W_{\mathcal{A}}(I')]$ found by the greedy algorithm.
2. Calculate the expected optimal weight $\mathbb{E}[W_{OPT}(I')]$.
3. Show that the ratio $\mathbb{E}[W_{OPT}(I')]/\mathbb{E}[W_{\mathcal{A}}(I')]$ is now bounded by a small constant (i.e., it no longer depends on the c parameter from Part A).

Part C. General Smoothed Analysis

Now, generalize your finding from Part B. For *any* n-job instance I, prove that the expected approximation ratio of Algorithm $\mathcal{A}$ on the perturbed instance I' is bounded by a function that is polynomial in n and $1/\sigma$. That is, show:

$$\mathbb{E}\left[\frac{W_{OPT}(I')}{W_{\mathcal{A}}(I')}\right] \leq \text{poly}\left(n, \frac{1}{\sigma}\right)$$

(Hint: You may need to "charge" the expected weight of any job k that is in OPT but not in $S_{\mathcal{A}}$ to the weight of the jobs in $S_{\mathcal{A}}$ that have a higher density than k.)