CSE 632: Analysis of Algorithms II: Randomized Algorithms — Fall 2019

Homework 5

Lecturer: Shi Li — **Deadline: Dec 3, 2019**

Problem 1 (10 points). Consider the following payoff matrix for a two-player 0-sum game, where the row player has two strategies a and b, and the column players have three strategies A, B and C

$$\begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 30 & 40 & 5 \\ 60 & 20 & 70 \end{pmatrix} \end{array}$$

The row player wants to minimize payoff value while the column player wants to maximize it. Give the value of the game and the optimum mixed strategies for both players.

Problem 2 (25 points). Let $\mathcal{S}$ be a family of subsets of $[n]$. Let $\mathcal{P} = \{w \in [0,1]^n : \sum_{i=1}^n w_i = 1\}$ be the $(n-1)$-dimensional simplex. Let $\alpha \in (0,1)$ be some number such that for every $w \in \mathcal{P}$, there exists some $S \in \mathcal{S}$ with $\sum_{i \in S} w_i \leq \alpha$.

Use the min-max theorem for 0-sum games to prove that there exists a distribution π over subsets of $\mathcal{S}$, such that for every $i \in [n]$ we have

$$\Pr_{S \sim \pi}[i \in S] \leq \alpha.$$

Moreover, assume there is a polynomial-time algorithm $\mathcal{A}$, that given any $w \in \mathcal{P}$, can find an $S \in \mathcal{S}$ with $\sum_{i \in S} w_i \leq \alpha$. Show that how to use $\mathcal{A}$ as a black box to find in polynomial time a distribution π' over $\mathcal{S}$ such that

$$\Pr_{S \sim \pi'}[i \in S] \leq \alpha + \epsilon, \forall i \in [n],$$

for an arbitrarily given constant $\epsilon > 0$.

Problem 3 (15 points). Given a metric space (X, d) with $|X| = n = ks$ for two integers $k, s \geq 1$, the balanced k-partition problem asks for a partitioning of X into k disjoint subsets $X_1, X_2, \cdots, X_k$ such that $|X_1| = |X_2| = \cdots = |X_k| = s$, so as to minimize

$$\sum_{i=1}^{k} \sum_{u \in X_i, v \in X_i} d(u, v).$$

Assume if (X, d) is a tree metric, there is a polynomial time $O(1)$-approximation algorithm for the problem: that is, the algorithm always produce a solution whose cost is at most $O(1)$ times the cost of the optimum solution. Show that for any metric (X, d), there is a (possibly randomized) polynomial time $O(\log n)$-approximation algorithm.