CSE 632: Analysis of Algorithms II | Fall 2017

Homework 2

Lecturer: Shi Li | **Deadline: 10/25/2017**

Instructions and Policy. Each student should write their own solutions independently and needs to indicate the names of the people you discussed a problem with. It is highly recommended that you think about each problem on your own for enough time before you discuss with others. It is recommended that you type your solutions using latex and submit the pdf file via UBLearns.

Problem 0 (0 points) You do not need to write down the solution for this problem. For a fixed $t \in \mathbb{R}$, e^{tx} is a convex function of x. By Jenson's inequality, we have $e^{tx} \leq 1 - x + xe^t = 1 + (e^t - 1)x$ for every $x \in [0, 1]$. Prove the Chernoff bounds we discussed in class, with the modification that each random variable x_i can take values in $[0, 1]$ (as opposed to $\{0, 1\}$).

Problem 1 (20 points) Consider the following linear programming rounding algorithm for the set cover problem. We solve the linear programming relaxation for the set cover problem as we discussed in the class, to obtain an optimum solution $x = (x_i)_{i \in [m]} \in [0, 1]^m$. Let $\alpha = c \ln n$ for a large enough constant $c > 0$ and let $x'_i = \min\{\alpha x_i, 1\}$ for every $i \in [m]$. For every $i \in [m]$, we choose the set S_i with probability x'_i (and do not choose i with probability $1 - x'_i$), independent of the choices for other sets. Prove that this algorithm gives an $O(\log n)$-approximation for the set cover, by proving the following statements:

(a) The expected cost of the solution we output is $O(\log n) \sum_{i=1}^m x_i$.

(b) When c is large enough, the probability that the solution we output is a valid set cover is at least $2/3$.

Problem 2 (40 points) In this problem, we consider a more general version of the machine minimization problem in which jobs have arrival times and deadlines. In the problem, we are given a set $[n]$ of jobs, where each job $j \in [n]$ has length p_j, an arrival time r_j and deadline d_j. So, we must process j on some machine during the time interval $(t, t + p_j) \subseteq (r_j, d_j)$, for some number t. The goal of the problem is use the minimum number of machines to process all the n jobs (of course, every machine can process at most 1 job at any time).

We assume all job sizes p_j and deadlines are integers in $[T]$, arrival times are integers in $[0, T - 1]$. For every job $j \in [n]$, we have $d_j \geq r_j + p_j$ and thus there is always a feasible solution. We also assume T is polynomial in n. That is, $T \leq n^c$ for some constant c.

Design an $O(\log n / \log\log n)$-approximation algorithm for this problem using linear programming relaxation and rounding.

Problem 3 (40 points) Show that the following linear programming is a linear programming relaxation for the congestion minimization problem. Moreover, every valid solution x to the LP relaxation leads to a valid LP solution $x' = (x'_p)_{p \in \bigcup_{i \in [k]} \mathbf{P}_i}$ to the LP relaxation we used in class, such that the congestion of x' is at most the congestion of x.

Our input graph $G = (V, E)$ is a directed graph. For every $v \in V$, we use $\delta^+(v)$ to denote the set of outgoing edges of v and $\delta^-(v)$ to denote the set of incoming edges of v.

$$\min \quad C \quad \text{s.t.}$$

$$\begin{aligned}
\sum_{e \in \delta^+(v)} x_{i,e} &= \sum_{e \in \delta^-(v)} x_{i,e} && \forall i \in [k], v \in V \setminus \{s_i, t_i\} \\
\sum_{e \in \delta^+(s_i)} x_{i,e} &= 1 && \forall i \in [k] \\
\sum_{i \in [k]} x_{i,e} &\leq C && \forall e \in E \\
x_{i,e} &\geq 0 && \forall i \in [k], e \in E
\end{aligned}$$