CMSC651 Midterm 1

Fan Yang

March 2018

Problem 1

Let's define a new random variable $Z = \frac{1}{Y}$. Since X and Y are independent, X and Z are independent as well.

In lecture 7 of *randlecnotes* we know the following property:

If X and Y are independent random variables, then $E[XY] = E[X] \cdot E[Y]$

Therefore we have:

$$E[\tfrac{X}{Y}] = E[XZ] = E[X] \cdot E[Z] = E[X] \cdot E[\tfrac{1}{Y}] = ac$$

Problem 2

Let X_i be the random variable for the event that the ith element is a fixed point in a random permutation.

$$X_i = \begin{cases} 1 & \text{if the } i\text{th point happens to be a fixed point} \\ 0 & \text{otherwise.} \end{cases}$$

We know $E[X_i] = Pr[X_i = 1] = \frac{1}{n}$. By linearity of expectation, the expected number of fixed points in a random permutation is:

$$E[X] = \sum_{i=1}^{n} E[X_i] = n \cdot \frac{1}{n} = 1$$

By definition of expectation we also know that $E[X] = \sum_{k=0}^{n} k \cdot \Pr[X = k]$, where $\Pr[X = k] = \frac{p_n(k)}{n!}$. Thus we have:

$$1 = \sum_{k=0}^{n} k \cdot \frac{p_n(k)}{n!}$$

which can be rewritten as:

$$n! = \sum_{k=0}^{n} k \cdot p_n(k)$$

Problem 3

Note: this answer is mainly adapted from chapter 1.7 of Shmoys-Williamson.

Define two LP relaxations for the multi-objective cover problem as below:

Problem 1 (P1):

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} w_j x_j \\ \text{subject to} \quad & \sum_{j:e_i \in S_j} x_j \geq 1, \qquad i = 1, \cdots, n \\ & x_j \geq 0, \qquad j = 1, \cdots, m \end{aligned}$$

Problem 2 (P2):

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} v_j x_j \\ \text{subject to} \quad & \sum_{j:e_i \in S_j} x_j \geq 1, \qquad i = 1, \cdots, n \\ & x_j \geq 0, \qquad j = 1, \cdots, m \end{aligned}$$

These two problems can be solved in polynomial time. Let x^* and $x^\#$ be the optimal solutions to P1 an P2 respectively, and let Z^*_{IP} and $Z^\#_{IP}$ be the their values. Given x^* and $x^\#$, the technique of *randomized rounding* can be applied to obtain a solution C to the integer program.

Imagine there are m biased coins $c^*_1, \cdots, c^*_m$ whose probabilities of coming up heads are $x^*_j, \cdots x^*_j$. We flip each of the coins $c \ln n$ times and include the corresponding set S_j in the solution if at least one of the flips turns out to be a head. Let P denote the solution constructed this way. We repeat the same procedures with m other coins whose whose probabilities of coming up heads are $x^\#_j, \cdots x^\#_j$, and name the yielded solution Q

Now let C be the intersection of P and Q, i.e., $C = P \cap Q$. We calculate the probability that a given element e_i is not covered in C:

$$\begin{aligned} \Pr[e_i \text{ not covered in } C] &\leq \Pr[e_i \text{ not covered in } P] + \Pr[e_i \text{ not covered in } Q] \\ &= \prod_{j:e_i \in S_j} (1 - x^*_j)^{c \ln n} + \prod_{j:e_i \in S_j} (1 - x^\#_j)^{c \ln n} \\ &\leq \prod_{j:e_i \in S_j} e^{-x^*_j (c \ln n)} + \prod_{j:e_i \in S_j} e^{x^\#_j (c \ln n)} \\ &= e^{-(c \ln n) \sum_{j:e_i \in S_j} x^*_j} + e^{-(c \ln n) \sum_{j:e_i \in S_j} x^\#_j} \\ &\leq \frac{1}{n^c} + \frac{1}{n^c} = \frac{2}{n^c} \end{aligned}$$

So the probability that C is a set cover is:

$$\Pr[C \text{ is a set cover}] \geq 1 - \sum_{i=1}^{n} \Pr[e_i \text{ not covered in } C] \geq 1 - \frac{2}{n^c} \cdot n = 1 - \frac{2}{n^{c-1}}$$

We now only need to prove that the expected value of C is bounded by $3W \ln n$ and $3V \ln n$ given that C is a set cover. Let $p_j(x^*_j)$ be the probability that a given subset S_j is included in P as a function of x^*_j. Since the probability of getting a head at each flip of the coin c^*_j is x^*_j, the probability of getting at least one head with $c \ln n$ flips is bounded by $(c \ln n) x^*_j$, *i.e.*, $p_j(x^*_j) \leq (c \ln n) x^*_j$. In fact, we have $0 \leq p_j(x^*_j) \leq \min(1, c \ln n)$ because $p_j(x^*_j) \in [0, 1]$. Similarly, we can derive $0 \leq p_j(x^\#_j) \leq \min(1, c \ln n)$.

Let X_j be a random variable that is 1 if the subset S_j is included in C, and 0 otherwise. Then:

$$\Pr[X_j = 1] = p_j(x^*_j) \cdot p_j(x^\#_j)$$

So the expectation of the weight function $\sum_{j\in C} w_j$ is:

$$E\left[\sum_{j\in C} w_j\right] = E\left[\sum_{j=1}^{m} w_j X_j\right] = \sum_{j=1}^{m} w_j \Pr[X_j = 1] = \sum_{j=1}^{m} w_j \cdot p_j(x_j^*) \cdot p_j(x_j^{\#})$$
$$\leq \sum_{j=1}^{m} w_j \cdot p_j(x_j^*) \leq \sum_{j=1}^{m} w_j \cdot (c\ln n)x_j^* = (c\ln n)\sum_{j=1}^{m} w_j x_j^* = (c\ln n)Z_{LP}^* \leq (c\ln n)W$$

$E\left[\sum_{j\in C} v_j\right] \leq (c\ln n)V$ can be proved by the same procedure.

Let F be the event that C is a set cover, and let $\bar{F}$ be the complement of it. In the previous discussion we have shown that $\Pr[F] \geq 1 - \frac{2}{n^{c-1}}$. We also know:

$$E\left[\sum_{j\in C} w_j\right] = E\left[\sum_{j\in C} w_j \middle| F\right]\Pr[F] + E\left[\sum_{j\in C} w_j \middle| \bar{F}\right]\Pr[\bar{F}]$$

Thus

$$\begin{aligned}
E\left[\sum_{j\in C} w_j \middle| F\right] &= \frac{1}{\Pr[F]}\left(E\left[\sum_{j\in C} w_j\right] - E\left[\sum_{j\in C} w_j \middle| \bar{F}\right]\Pr[\bar{F}]\right) \\
&\leq \frac{1}{\Pr[F]} \cdot E\left[\sum_{j\in C} w_j\right] \\
&\leq \frac{(c\ln n)W}{1 - \frac{2}{n^{c-1}}} \\
&\leq 3W\ln n \quad (\text{if } c = 2 \text{ and } n \geq 6)
\end{aligned}$$

Similarly, we can prove $E[\sum_{j\in C} v_j|F] \leq 3V\ln n$ when $c = 2$ and $n \geq 6$.

It's obvious that the algorithm runs in polynomial time (solve two LP relaxations and flip coins $\mathcal{O}(cm\ln n)$ times).

Problem 4

Let $i = 0, 1, \cdots$ be the indices of the intervals, and let a_i be the left-end point of the ith interval and b_i be the length of the $i$the interval. A better rounding-down scheme is to make $a_0 = T/k$ and $b_i = \frac{T}{k^2}(1+\frac{1}{k})^i$. Note that we can make $a_0 = T/k$ because all long jobs have length greater than T/k. We will now show that the total number of intervals is $\mathcal{O}(k\ln k)$.

For any a_i, we have:

$$a_i = a_0 + \sum_{k=0}^{i-1} b_k = \frac{T}{K} + \sum_{k=1}^{i-1}\frac{T}{k^2}\left(1+\frac{1}{k}\right)^k = \frac{T}{K} + \frac{T}{k^2}\cdot\frac{(1+\frac{1}{k})^i - 1}{(1+\frac{1}{k}) - 1} = \frac{T}{k}\left(1+\frac{1}{k}\right)^i$$

Consider the largest a_l. We know:

$$\begin{aligned}
& a_l = \frac{T}{k}\left(1+\frac{1}{k}\right)^l < T \\
\Longrightarrow \quad & \left(1+\frac{1}{k}\right)^l < k \\
\Longrightarrow \quad & l < \log_{(1+\frac{1}{k})} k < \frac{\ln k}{\ln(1+\frac{1}{k})} = \frac{\ln k}{\frac{1}{k} - \frac{1}{2k^2} + \frac{1}{3k^3}\cdots} \approx k\ln k
\end{aligned}$$

Therefore, with this rounding-down strategy there are at most $n^{\mathcal{O}(k \ln k)}$ distinct inputs to the intervals. We then can use the same dynamic programming method as described in chapter 3.2 to determine whether or not there exists a feasible schedule that processes these rounded long jobs within time T. This dynamic programming subroutine finishes within time of $n^{\mathcal{O}(k \ln k)}$. Assume it finds a feasible schedule S for a given T for the rounded long jobs. We now prove that the schedule has a value of at most $(1 + \frac{1}{k})T$ when it's applied to the original long jobs.

We have a very simple observation, that is, for each interval i:

$$\frac{b_i}{a_i} = \frac{\frac{T}{k^2}\left(1 + \frac{1}{k}\right)^i}{\frac{T}{k}\left(1 + \frac{1}{k}\right)^i} = \frac{1}{k}$$

Hence, for each job $j \in S$, the difference between its true processing requirement and its rounded one is at most $1/k$. This leads to the following conclusion,

$$\sum_{j \in S} p_j \leq T + \frac{1}{k} \cdot T = \left(1 + \frac{1}{k}\right)T$$

Finally, we need to convert the above algorithm into a PTAS. Here the same bisection search procedure discussed for Theorem 3.7 can be adopted without modification. Below is a brief summary of it.

The optimal makespan for the scheduling input is within the interval $[L_0, U_0]$, where

$$L_0 = \max\left\{\left\lceil \sum_{j=1}^{n} p_j/m \right\rceil, \max_{j=1,\cdots,n} p_j\right\}$$

and

$$U_0 = \left\lceil \sum_{j=1}^{n} p_j/m \right\rceil + \max_{j=1,\cdots,n} p_j$$

In each iteration where the current interval is $[L, U]$, we set $T = \lfloor (L + U)/2 \rfloor$, and run the DP subroutine on the rounded long jobs, where $k = \lceil 1/\epsilon \rceil$. If the subroutine produces a schedule, then update $U \leftarrow T$; otherwise, update $L \leftarrow T + 1$. The bisection repeats polynomial times until $L = U$, at which point, the DP subroutine outputs the schedule associated with L. We know $C^*_{max} \geq L$, and that the final schedule output by the subroutine is at most $(1 + \epsilon)L \leq (1 + \epsilon)C^*_{max}$, so the algorithm is a PTAS.