CSL 356: Analysis and Design of Algorithms
Instructor: Ragesh Jaiswal

1. Discuss Homework-5

2. (*Weak Duality*) A linear programming problem in the standard form can be written in short using the following vector notation:

$$\boxed{\begin{aligned} \textbf{LP1}\text{: Maximize } (c^T \cdot x),\\ \text{subject to } A \cdot x \leq b\\ x \geq 0 \end{aligned}}$$

Here x, c, b are vectors and A is a $m \times n$ matrix. Consider the following related linear program:

$$\boxed{\begin{aligned} \textbf{LP2}\text{: Minimize } (b^T \cdot y),\\ \text{subject to } A^T \cdot y \geq c\\ y \geq 0 \end{aligned}}$$

Here y is vector of size m. Consider the following linear program:

Let f_1 be any feasible solution for LP1 and let f_2 be any feasible solution for LP2. Show that $f_1 \leq f_2$.

3. (*Randomized rounding*) Linear programming can be used to obtain approximation algorithms for NP-hard problems. Consider the Set cover problem. Here we are given subsets $S_1, ..., S_m \subseteq U$ and we want to find the minimum sized subset $I \subseteq \{S_1, ..., S_m\}$ such that the union of elements of subsets in S is equal to U. We may solve the following ILP to obtain a solution to the problem:

$$\boxed{\begin{array}{l} \text{Minimize } \sum_{i=1}^{m} x_i,\\ \text{Subject to }:\\ \sum_{i:e \in S_i} x_i \geq 1, \text{ for all } e \in U\\ x_i \in \{0, 1\}, \text{ for all } x_i \end{array}}$$

The issue here is that we know ILP is NP-hard. We relax the integer constraint to obtain the following LP:

$$\boxed{\begin{array}{l} \text{Minimize } \sum_{i=1}^{m} x_i,\\ \text{Subject to }:\\ \sum_{i:e \in S_i} x_i \geq 1, \text{ for all } e \in U\\ x_i \geq 0, \text{ for all } x_i \end{array}}$$

Suppose the optimal solution of the above LP be $x_1^*, ..., x_m^*$. Now we pick the set cover using the following randomized algorithm:

1. $I \leftarrow \phi$
2. For all i: $I \leftarrow I \cup S_i$ with probability x_i^*.
3. Repeat (2) until all elements are covered.

Show the following:

(a) The number of times step (2) is executed is at at most $(2 \log n)$ with probability at least $(1 - 1/n)$.

(b) The above randomized algorithm with high probability gives an $O(\log n)$ approximation to the set cover problem.