Assignment 2

Please type down your answers using Latex. Please hand in your assignment (by email or by hard copy) before the class of October 28, 2020. You are welcome to discuss among yourselves, but everyone must write down his/her own answers. Please feel free to contact me if you have questions.

Generalisation of Set Cover

Suppose that for each element e in the universe U, there is also a "demand" $r(e) \in \mathbb{Z}$, which means the number of times that this element e must be covered. Notice that you are allowed to use the same set S in the given family $\mathbb{S}$ of sets *multiple times* in your solution. Of course, if a set S is included t times, you must pay the cost of $t \cdot w(S)$. The goal is still to minimise the total cost.

Imitate the greedy algorithm and its analysis that we have presented in class, to design an approximation algorithm. Of course you should define the greedy properly in this context and prove its approximation ratio. For analysis, maybe you can start by writing down the primal and dual programs of this problem and go from there.

For the people who are perfectionists: there is a further generalisation. Suppose that each set S in the family $\mathbb{S}$ is a "multi-set", that is, each set S is specified with not only the elements it covers, but also the number of times that each element is covered by it. So assume that we are given $M(S, e) \in \mathbb{Z}_{\geq 0}$ to denote how many times e is covered by set $S \in \mathbb{S}$. Notice that you still have the right to add the same set S multiple times into your final solution. Could you generalise your greedy algorithm to this setting?

Application: you may wonder what is the point of these generalisations. Here is an application. We are given the following integer programming problem.

$$
\begin{aligned}
\min c^T x \\
Ax \geq b \\
x \in \mathbb{Z}^n_{\geq 0}
\end{aligned}
$$

It is easy to see that the above generalisation of set cover can be used to solve this problem.

Integrality Gap

An extremely important notion in linear programming is that of "integrality gap." To illustrate, consider a minimisation problem (think of set cover). Let F and F_Z denote the set of feasible fractional solutions and the set of integral feasible solutions. The integrality gap is then defined as

$$\frac{\min_{x \in F_Z} w(x)}{\min_{x \in F} w(x)}.$$

Argue that in the above generalisations of set cover, the approximation ratio you have coincides with the integrality gap.

Next, consider the second generalisation. Suppose that we impose the restriction that each set can be taken at most once (implying that in the application, the vector x can only take values of 0 or 1). Construct an example to show that the integrality gap can much larger than $\Omega(\log n)$.