1. Introduction to Approximation Algorithm

Bingyue Bei

October 25, 2024

Exericse 1.1

i For the maximization problem, we have definition:

$$\textsc{alg}(\mathrm{I}) \geq \rho \cdot \textsc{opt}(\mathrm{I}) \quad \forall \text{ I as input.}$$

If $\rho > 1$, we have

$$\textsc{alg}(\mathrm{I}) \geq \rho \cdot \textsc{opt}(\mathrm{I}) > \textsc{opt}(\mathrm{I}),$$

which contradicts the definition of an optimal solution.

ii For the minimization problem, we have definition:

$$\textsc{alg}(\mathrm{I}) \leq \rho \cdot \textsc{opt}(\mathrm{I}) \quad \forall \text{ I as input.}$$

If $\rho < 1$, we have

$$\textsc{alg}(\mathrm{I}) \leq \rho \cdot \textsc{opt}(\mathrm{I}) < \textsc{opt}(\mathrm{I}),$$

which contradicts the definition of an optimal solution.

Exercise 1.2

We say that the approximation ratio ρ is tight when:

$$\rho = \inf_{I} \frac{\textsc{alg}(\mathrm{I})}{\textsc{opt}(\mathrm{I})}$$

where the infimum is taken over all possible inputs I.

Exercise 1.3

i Consider the job array [80, 80, 40]. Note here that $\text{OPT(I)} = 80 + 40 = 120$. The work distribution is:

- Machine 1: [80, 40]
- Machine 2: [80]

In this case, Alg returns the optimal result, and is indeed a 1.05-approximation.

ii We started by building an intuition for the problem we are facing. Since the tasks all have very small sizes, in the optimal case, they will be distributed evenly between the two machines, returning a makespan which is approximately 100. So the result for a 1.05-approximation algorithm should be slightly above and very close to 105.
Here is a formal proof: Since $t_1, t_2, \ldots, t_n \leq 10$, let M_i be the workload on machine o_1 and M_2 be the workload on machine o_2. We have $|M_1 - M_2| \leq 20$ for an optimal solution.
We proof this claim by constructing a proof by contradiction. Without loss of generality, assume $M_1 \geq M_2 + 20$. Remove any task, represented by t', from the workload of M_1, and added it into the workload of M_2. The new workload for o_1, $M_1' = M_1 - t' \geq M_1 - 10$ since $t' \leq 10$. The new workload for o_2, $M_2' = M_2 + t' \leq M_2 + 10$. Since $M_1 \geq M_2 + 20$, we have $M_1 - 10 \geq M_2 + 10$, which is equivalent to $M_1' \geq M_2'$. The makespan have been reduced from M_1 to M_1' We thus created a new solution better than the previous one which claimed to be the optimal.
Since $M_1 + M_2 = 200$, and $|M_1 - M_2| \leq 20$. For an optimal solution, we have $M_1, M_2 \leq \frac{200+20}{2} = 110$. So the optimal solution has an upper-bound of 110, $\textsc{opt}(\text{I}) \leq 110$. Our algorithm is an $1.05 -$ approximation, meaning $\textsc{alg}(\text{I}) \leq 1.05 \cdot \textsc{opt}(\text{I}) = 1.05 \cdot 110 = 115.5$.
Professor's claim has to be false.

Exercise 1.4

A

B

Exercise 1.5

i Given the constraints of the problem we have:

$$\sum_{j=1}^{n} t_j \geq 500, \quad t_1, t_2, ..., t_n \leq 25$$

Let M_i be the machine whose workload determines the makespan. Consider the scheduling of its last job j^* with time t_{j^*}. Before that job get scheduled, the total processing time of all the job which has been scheduled, represented by $load(M_i^*)$ is at least 475.

$$load(M_i^*) = \frac{1}{5}(\sum_{j=1}^{j<j^*} t_j) \geq \frac{1}{5}(\sum_{j=1}^{n} t_j - t_{j^*}) \geq \frac{(500-25)}{5} = \frac{475}{5}$$

Since that machine is where the last job is assigned to, workload of that machine has an upper-bound of $95 + 25 = 120$.

ii Considering the task array of [1, 1, ..., 1, 25],