Lecture 2.2

Wednesday, 6 September 2023 21:23

Even if $OPT(I)$ is unknown, it is still possible to compute a value of ρ. This is typically done by finding a lower bound $LB(I)$ on the value of $OPT(I)$. Then, the inequality can be written as $ALG(I) \le \rho \cdot LB(I) \le \rho \cdot OPT(I)$ *(Note: this holds for minimization problems; for maximization problems, an upper bound is used instead.)*

For the load-balancing problem, two obvious lower bounds exist:
1. At least one machine has at least the average load of all machines.
2. Some machine must execute the job with maximum load.

Taken together, this can be written as follows:

$$OPT(I) \ge \max\left(\frac{1}{m} \cdot \sum_{j=1}^{n} t_j \,,\ \max_{1 \le j \le n} t_j\right) = LB(I)$$

Proofs of statements saying that an algorithm ALG is a ρ-approximation algorithm are roughly structured as follows:
$ALG(I) \le \cdots \le \rho \cdot LB(I) \le \rho \cdot OPT(I)$

For the proof, we use the following definitions:
- M_{i^*} is the machine with the largest load, i.e. the machine which determines the makespan;
- J_{j^*} is the last job assigned to M_{i^*}.
- $Load^*(M_{i^*})$ is the load on M_{i^*} just before J_{j^*} was assigned.

We have that

$$\text{Greedy-Scheduling}(I) = Load^*(M_i^*) + t_{j^*}$$

$$\le \boxed{\frac{1}{m} \cdot \sum_{j=1}^{j^*-1} t_j} + t_{j^*}$$

note: since job j^* is assigned to the machine with minimal load before adding j^*, the load of that machine is at most the average load over the first j^*-1 jobs.

$$\boxed{\le} \frac{1}{m} \sum_{j=1}^{n} t_j + t_{j^*}$$

note: since $n \ge j^*-1$, and all t_j's are positive, this sum is greater or equal to the previous one

$$\le \boxed{2} \cdot LB(I) \le 2 \cdot OPT(I)$$

to obtain this value, we assume that both terms in the max are equal

$$\le \frac{1}{m}\left(\left(\Sigma_{j=1}^{n} t_j\right) - t_{j^*}\right) + t_{j^*}$$

$$\le \frac{1}{m}\left(\Sigma_{j=1}^{n} t_j\right) + \left(1 - \frac{1}{m}\right) \cdot t_{j^*}$$

$$\le \left(2 - \frac{1}{m}\right) LB(I) \le \left(2 - \frac{1}{m}\right) OPT(I)$$

From this proof, it follows that Greedy-Scheduling is a $\left(2 - \frac{1}{m}\right)$-approximation algorithm.

Note that the higher the number of machines, the worse the approximation ratio will get *(i.e. it will get closer to two)*.

It should also be noted that this bound is tight; for any m, there are inputs such that $Greedy - Scheduling(I) \le \left(2 - \frac{1}{m}\right) \cdot OPT(I)$.

There are, in general, three strategies for improving the approximation ratio for a given problem:
1. Use the same algorithm, but perform a better analysis using the same lower bound (such that ρ becomes lower);
2. Use the same algorithm, but perform a better analysis using a different lower bound;
3. Construct a new algorithm (possible using a different lower bound).