Lecture 2.3

Thursday, 7 September 2023 09:30

idea to improve greedy-scheduling:

1. first, sort the jobs from largest to smallest
2. second, run greedy-scheduling on the sorted input

Theorem ordered-scheduling is a $\frac{3}{2}$-approximation algorithm

assume there are at least $m+1$ jobs; otherwise, every job can get its own machine, leading to a trivially optimal solution

some machine has load at least $t_m + t_{m+1}$, since the machine with job m will have the lowest load after being assigned job m and hence be assigned job $m+1$ as well

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

Let I be an arbitrary input instance

case 1: $j^* \le m$. Then job j^* gets its own machine, since all of the first m jobs can get their own machine.

Then, the makespan must be optimal, since it is impossible to reduce the load w/o the job determining the makespan is already running on its own machine.

case 2: $j^* \ge m+1$. Then,

$$\text{ordered-scheduling}(I) \le Load^*(M_{i^*}) + t_{j^*}$$

$t_{j^*} \le t_{m+1} \le \frac{1}{2}(t_m + t_{m+1})$ because $j^* \ge m+1$ (and the jobs are sorted)

$$\le \frac{1}{m}\sum_{j=1}^{j^*-1} t_j + \frac{1}{2}(t_m + t_{m+1})$$

$$\le \frac{1}{m}\sum_{j=1}^{n} t_j + \frac{1}{2}(t_m + t_{m+1})$$

$$\le \frac{3}{2}\cdot LB(I) \le \frac{3}{2}\cdot OPT(I)$$

note: both of these components cannot be greater than the maximum in the lower bound

otherwise, they would determine the value of the maximum, leading to a contradiction

It is also possible to prove a lower bound of $\frac{3}{2} - \frac{1}{2m}$.

The modification of the proof is similar to the one used in the previous lecture for greedy-scheduling.