# Makespan minimization

$m$ machines
job $1, \ldots, n$

**Input**: An integer $m > 0$, and $n$ integers $a_1, \ldots, a_n > 0$.

We need to assign each job to a unique machine.

Makespan of machine $j$ = $\sum_{\substack{\text{jobs } i \\ \text{assigned to} \\ \text{machine } j}} a_i$
$M(j)$
(w.r.t. a schedule)

Makespan of a schedule $= \max_{j=1}^{m} M(j)$.

**Task**: Find a schedule with minimum makespan.

NP-hard even for $m = 2$.

## Greedy algorithms:

- Consider the jobs in any order
- At each step, assign a job to a least loaded machine.

## Lower bounding schemes

$$\text{OPT} \geq \max_{i=1}^{n} a_i \qquad \text{— (1)}$$

$$\text{OPT} \geq \frac{\sum_{i=1}^{m} a_i}{m} \qquad \text{— (2)}$$

$$\left[ \sum_{j=1}^{m} M(j) = \sum_{j=1}^{n} a_i \Rightarrow \text{OPT} \geq \frac{\sum M(j)}{m} = \frac{\sum a_i}{m} \right]$$

**Theorem 1:** The greedy algorithm achieves an approximation factor of 2.

**Proof:** Let Machine $j$ has maximum makespan. Let $i$ be the job that is assigned to it last. Just before $i$ was assigned to it, its makespan was at most $\sum_{k=1}^{n} a_k / m$ (in fact it is at most $\dfrac{\sum_{k=1}^{n} a_k - a_i}{m}$).

$$\text{Thus, makespan of the computed sol}^n = M(j) \le \frac{1}{m}\sum_{k=1}^{n} a_k + a_i$$

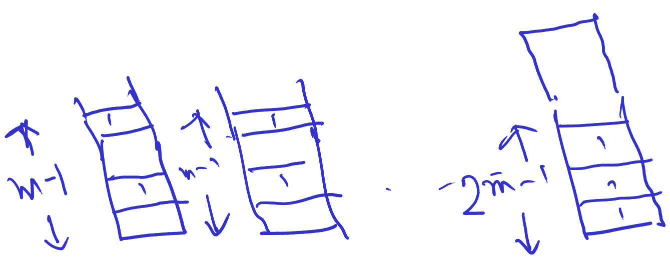
$$\le OPT + OPT \quad (\text{by } 1 \ \& \ 2)$$

$$= 2 \cdot OPT.$$

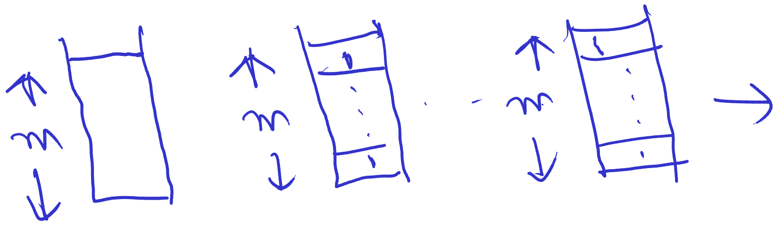
$\hookrightarrow$ improvable to $2 - \frac{1}{m}$.

$\blacksquare$

The bound of Theorem 1 is tight. Consider the following sequence of jobs: $\underbrace{1, 1, \ldots\ldots, 1}_{m(m-1)}, m$.




$\rightarrow$ sol$^n$ computed by the greedy.
makespan $= 2m-1$.



$\rightarrow$ OPT $= m$. Ratio $= 2 - \frac{1}{m}$.

## Modified greedy

- Consider the jobs in non-increasing order of processing times. Let that sequence be $a_1, a_2, \ldots, a_n$.
- In each iteration, assign a job to a least loaded machine.

**Theorem 2**: The modified greedy algorithm is a $\frac{3}{2} - o(1)$ approximation algorithm.

**Proof**: Let $M(j)$ be the makespan of the computed schedule, and let $i$ be the last job that is assigned to $j$.

**Case-1**: $i \leq m$. The first $m$ jobs are assigned to distinct machines. So, $M(j) = a_i \leq \max_{k=1}^{n} a_k$
$\leq$ OPT.

$\therefore$ The computed solution is optimal.

**Case-2**: $i \geq m+1$. In any optimal schedule, there exist $1 \leq k < l \leq m+1$ such that jobs $k$

and $l$ are assigned to the same machine $p$ (by pigeon-hole principle).

$$OPT \geq M(p) \geq a_k + a_l \geq 2a_i$$

$$\Rightarrow a_i \leq \frac{1}{2} \cdot OPT.$$

Before $i$ was assigned to the $j$-th machine, the makespan of $j$ was at most $\sum_{q=1}^{n} a_q / m$.

$\leq OPT$ (by 2)

$$\therefore M(j) \leq OPT + \frac{1}{2} \cdot OPT = \frac{3}{2} \cdot OPT \qquad \blacksquare$$

[ M( ) depends on the schedule, which is suppressed in our notation]