# Online Parallel Job Scheduling Special Cases

Jacob Jan Paulus

and

Johann L. Hurink

.

j.j.paulus@utwente.nl

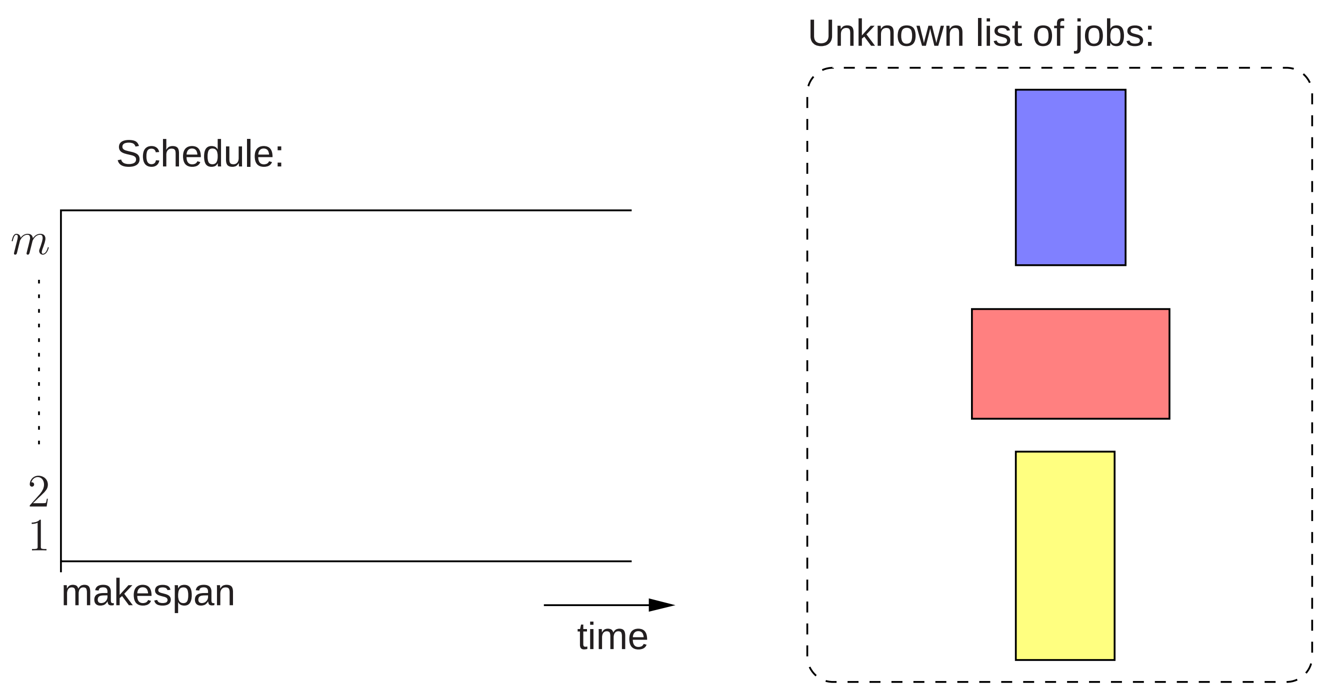
Department of Applied Mathematics

University of Twente

# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: ($P|\mathrm{online}-\mathrm{list}, m_j|C_{\max}$)

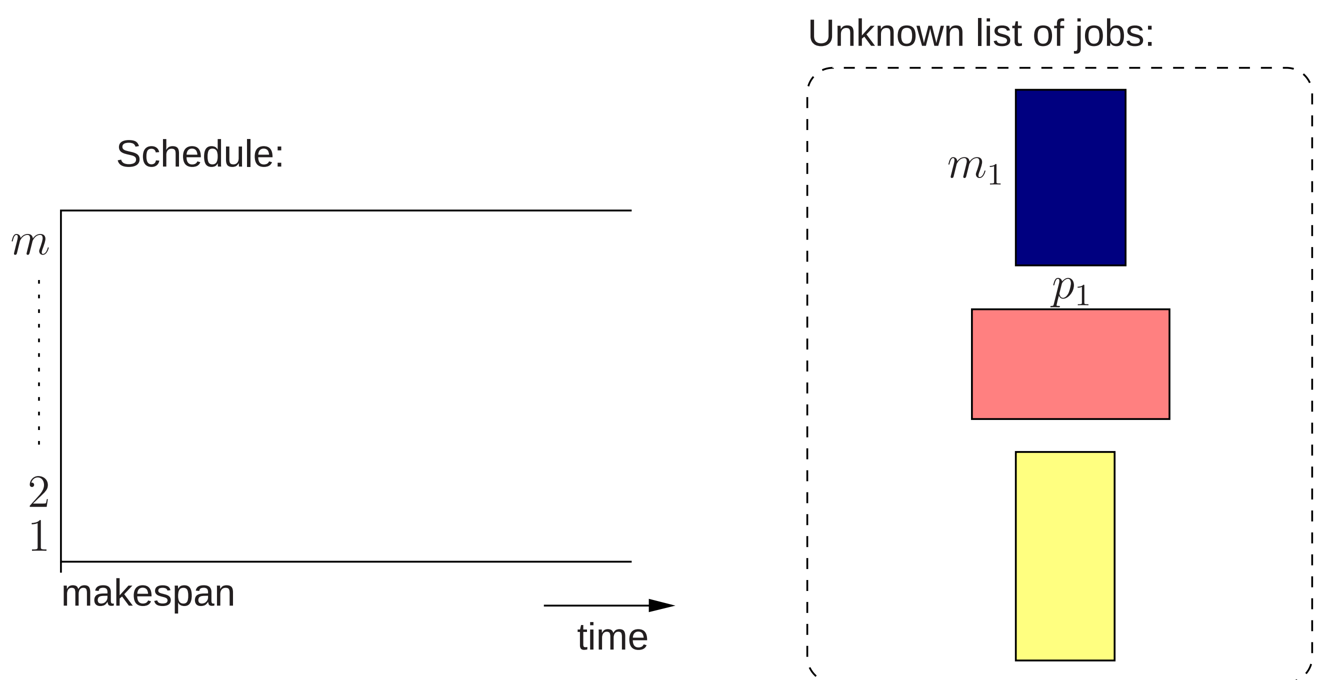
- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: ($P|\mathrm{online}-\mathrm{list}, m_j|C_{\max}$)

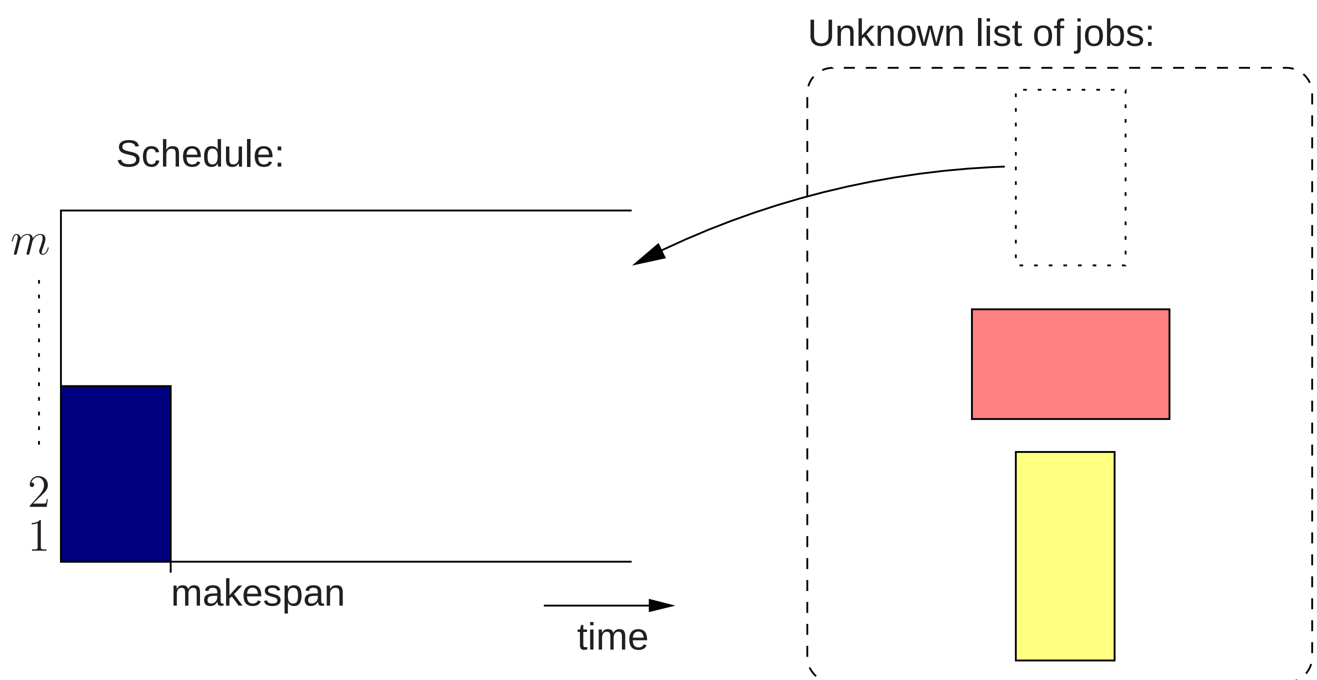
- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: ($P|\mathrm{online-list}, m_j|C_{\max}$)

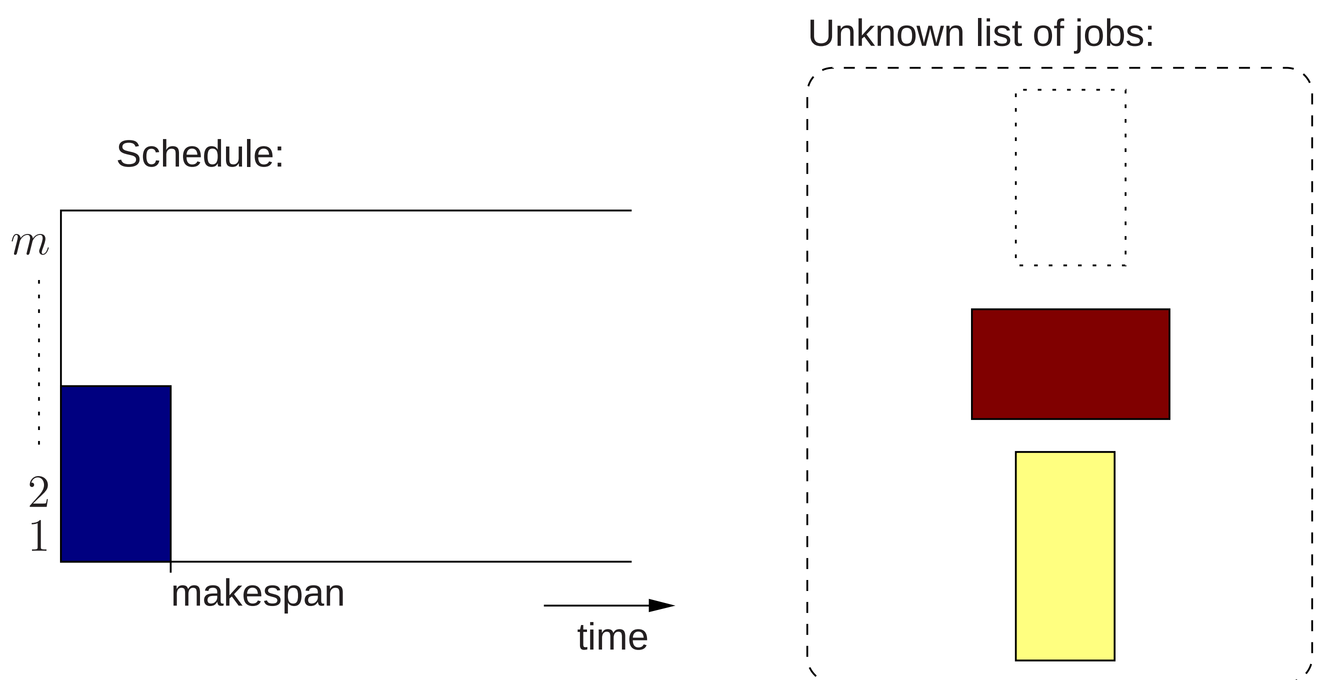
- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: ($P|\mathrm{online} - \mathrm{list}, m_j|C_{\max}$)

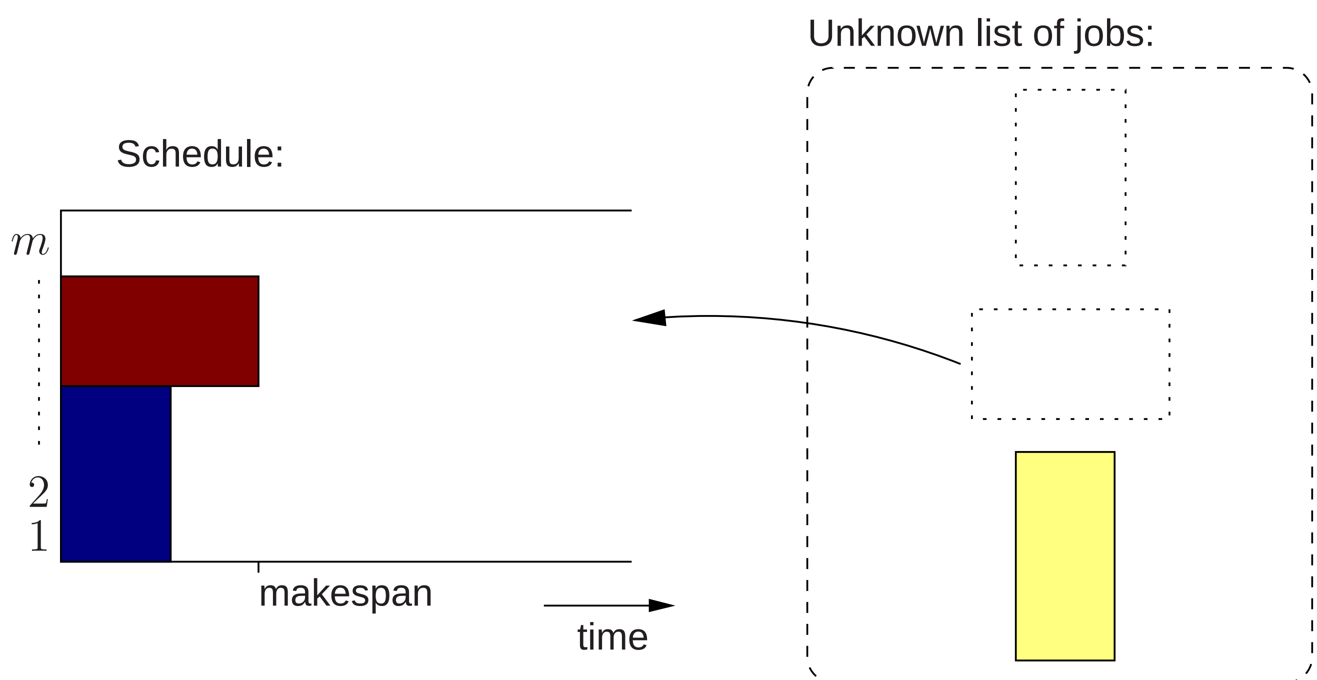
- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: $(P|\mathrm{online}-\mathrm{list}, m_j|C_{\max})$

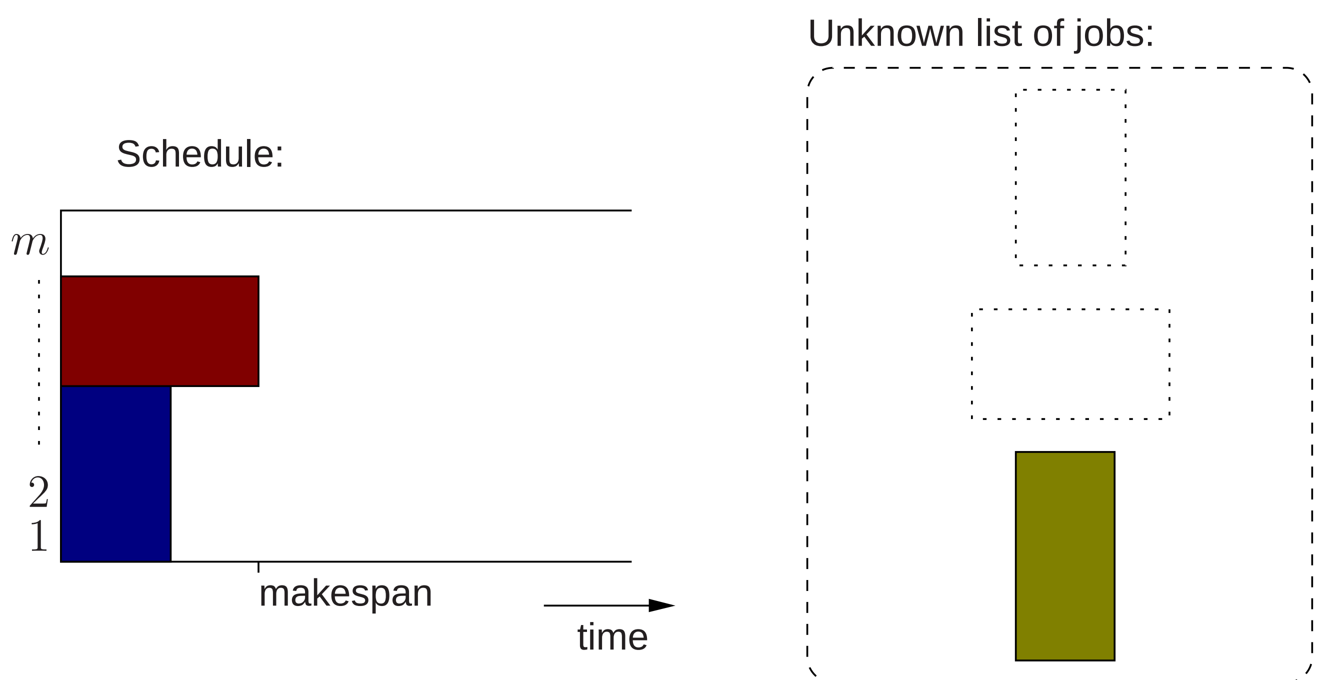
- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: ($P|\mathrm{online}-\mathrm{list}, m_j|C_{\max}$)

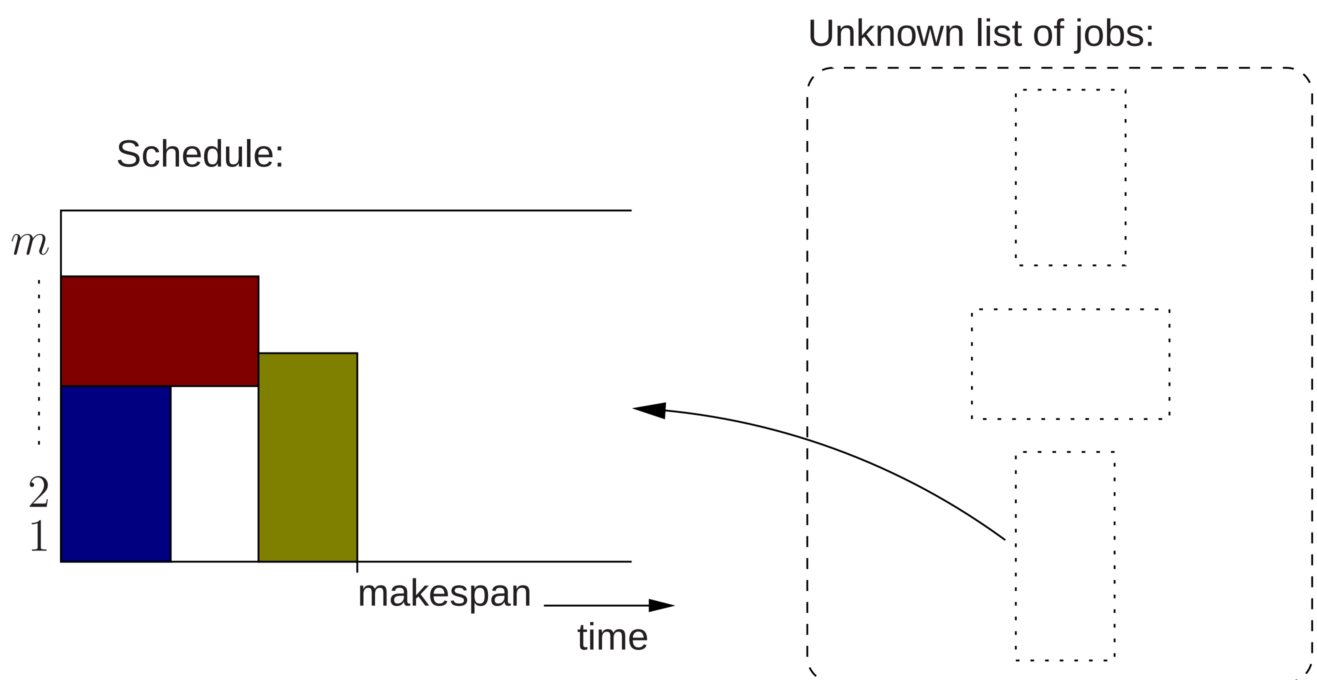
- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: ($P|\mathrm{online}-\mathrm{list}, m_j|C_{\max}$)

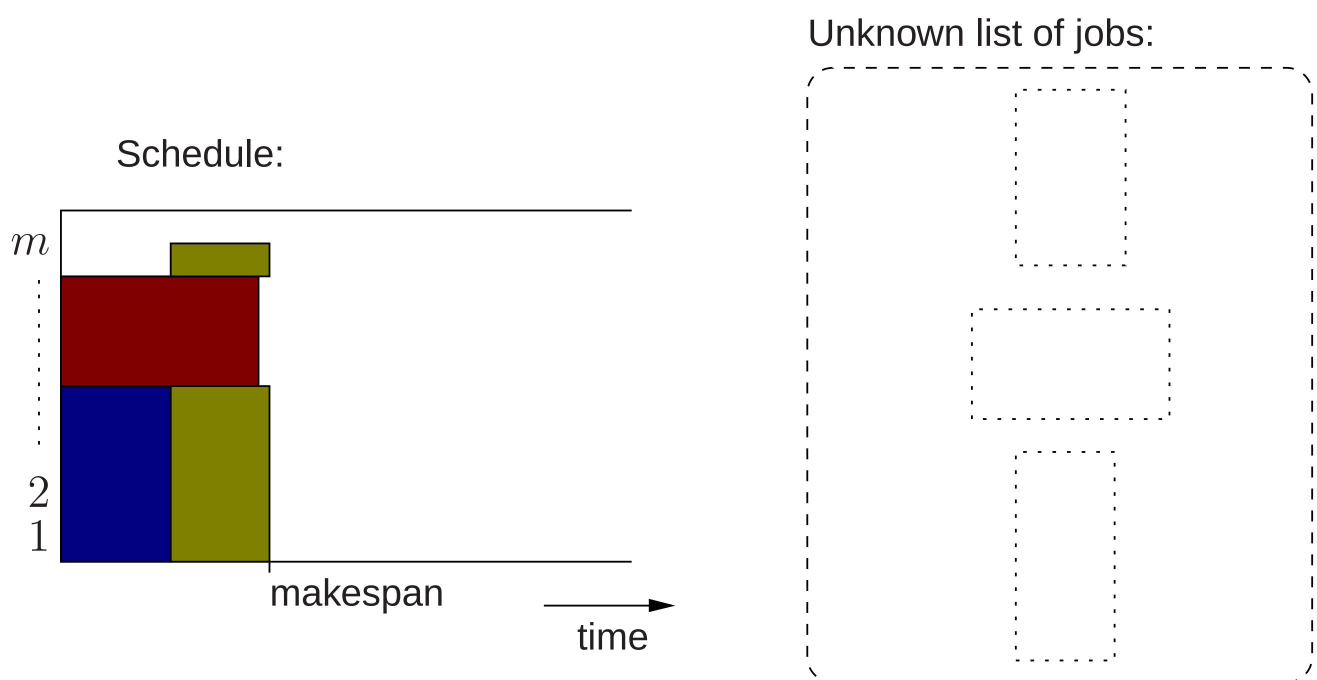
- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Parallel Job Scheduling*

Scheduling parallel jobs on $m$ machines: $(P|\mathrm{online} - \mathrm{list}, m_j|C_{\max})$

- Jobs have a processing time ($p_j$) and a number of machines simultaneously required for processing ($m_j$),
- As soon as a job arrives, it has to be scheduled irrevocably without knowing the characteristics of the future jobs,
- Preemption is not allowed,
- The objective is to minimize the makespan.



# *Online Algorithms: The Analysis is a Game*

An online algorithm $A$ is said to be $\rho$-competitive if

$$\sup_{\sigma} \frac{C_A(\sigma)}{C^*(\sigma)} \leq \rho \ ,$$

where $C^*$ is the value of the optimal offline solution.

Interpret the analysis as a game between the online algorithm and an adversary.

- Online algorithm schedules the jobs to minimizes the competitive ratio.
- Adversary determines the next job characteristics to maximize the competitive ratio.

# *Online Algorithms: The Analysis is a Game*

An online algorithm $A$ is said to be $\rho$-competitive if

$$\sup_{\sigma} \frac{C_A(\sigma)}{C^*(\sigma)} \leq \rho \ ,$$

where $C^*$ is the value of the optimal offline solution.

Interpret the analysis as a game between the online algorithm and an adversary.

- Online algorithm schedules the jobs to minimizes the competitive ratio.
- Adversary determines the next job characteristics to maximize the competitive ratio.

To show

- a lower bound on $\rho$: Construct an adversary and show that no online algorithm can be better than $\rho$-competitive.
- a upper bound on $\rho$: Construct an online algorithm and show that matter what the adversary does, it is $\rho$ competitive.

# *Known Results*

| $P\|\text{online}-\text{list}, m_j\|C_{\max}$ | | | |
|---|---|---|---|
| Model | Lower Bound | Upper Bound | |
| - | $2.43$ | $6.6623$ | |
| $m = 2$ | $2$ | $2$ | *Greedy* |
| $m = 3$ | $2$ | $2.8$ | *This talk* |
| $3 \leq m \leq 6$ | $2$ | $m$ | *Greedy* |
| Semi-online $P\|online-list, m_j\|C_{max}$ | | | |
| Model | Lower Bound | Upper Bound | |
| -non-increasing $m_j$ | $1.88$ | $2.4815$ | *This talk* |
| $m = 2$ or $3$ | $2 - \frac{1}{m}$ | $2 - \frac{1}{m}$ | *Greedy* |
| $m = 4$ or $5$ | - | $2$ | *Greedy* |
| -non-increasing $p_j$ | $\frac{5}{3}$ | $2$ | |
| $m = 2$ | $\frac{9}{7}$ | $\frac{4}{3}$ | |
| -non-decreasing $p_j$ | - | - | |
| $m = 2$ | $\frac{3}{2}$ | $\frac{3}{2}$ | |

# *Overview*

- Lower bounds on optimal solutions
- Greedy
- Case: $m = 3$
- Case: non-increasing $m_j$

# *Lower bounds on optimal solutions*

Given a list of jobs $\sigma$

- Load argument:

$$C^*(\sigma) \geq \frac{1}{m} \sum_{j \in \sigma} m_j p_j$$

- Length argument:

$$C^*(\sigma) \geq \max_{j \in \sigma} \{p_j\}$$

# *The Greedy Algorithm*

- Greedy is $m$-competitive.

*Proof:* In a schedule constructed by Greedy never $m$ machines are left idle. By the load argument we get

$$\begin{aligned} C_{\text{Greedy}}(\sigma) &\leq \sum_{j\in\sigma} m_j p_j \\ &\leq mC^*(\sigma) \end{aligned}$$

So,

$$\frac{C_{\text{Greedy}}(\sigma)}{C^*(\sigma)} \leq m$$

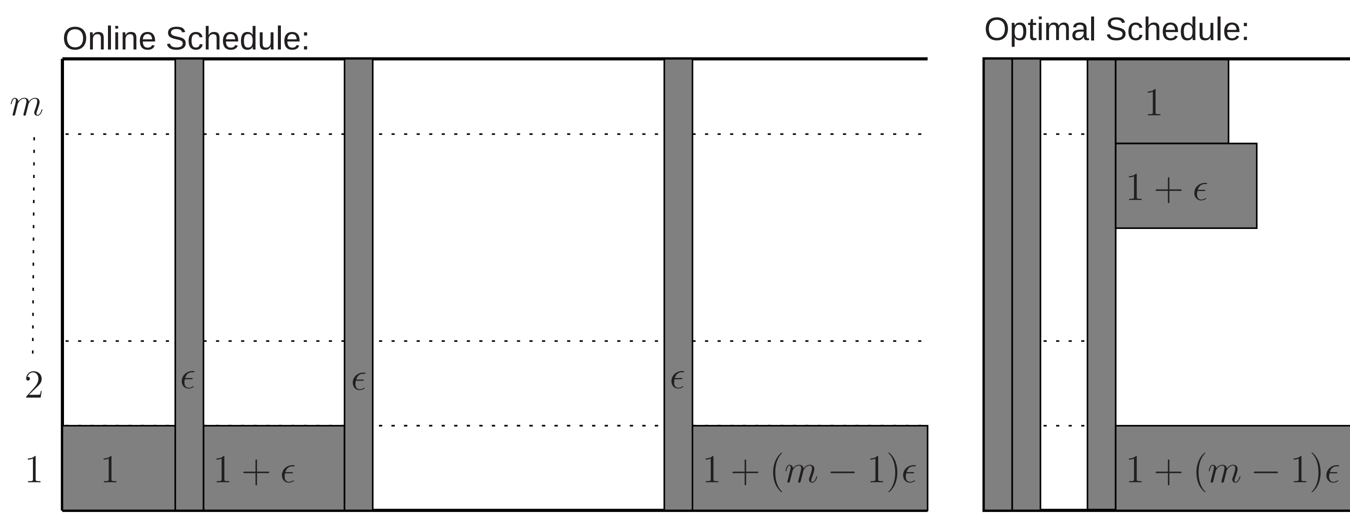
# *The Greedy Algorithm*

- Greedy is $m$-competitive.

*Proof:*
Consider the illustrated instance.

$$\frac{C_{\text{Greedy}}(\sigma)}{C^*(\sigma)} = \frac{\sum_{i=1}^{m}(1+(i-1)\epsilon)+(m-1)\epsilon}{1+(2m-2)\epsilon}$$

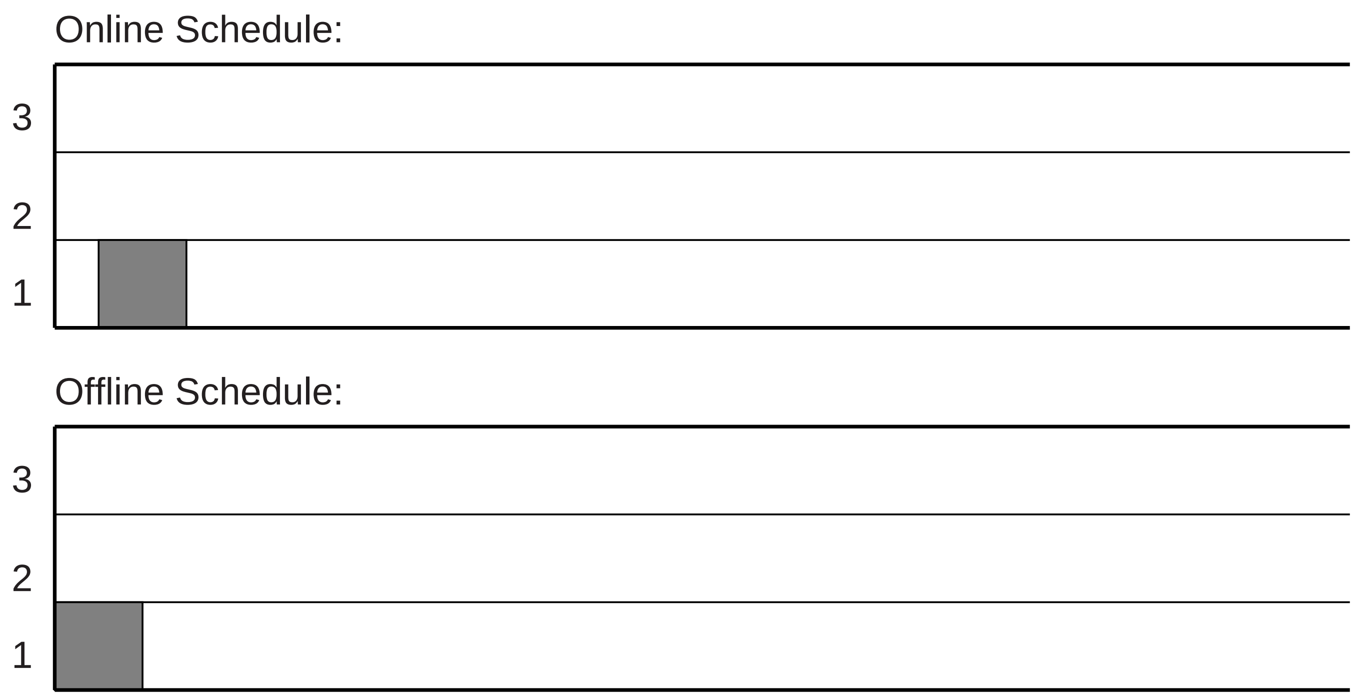
$$= \frac{\frac{1}{2}\epsilon m(m-1)+m-\epsilon}{1+(2m-2)\epsilon} \rightarrow m \text{ if } \epsilon \rightarrow 0$$



# *Case:* $m = 3$

Lower bound:

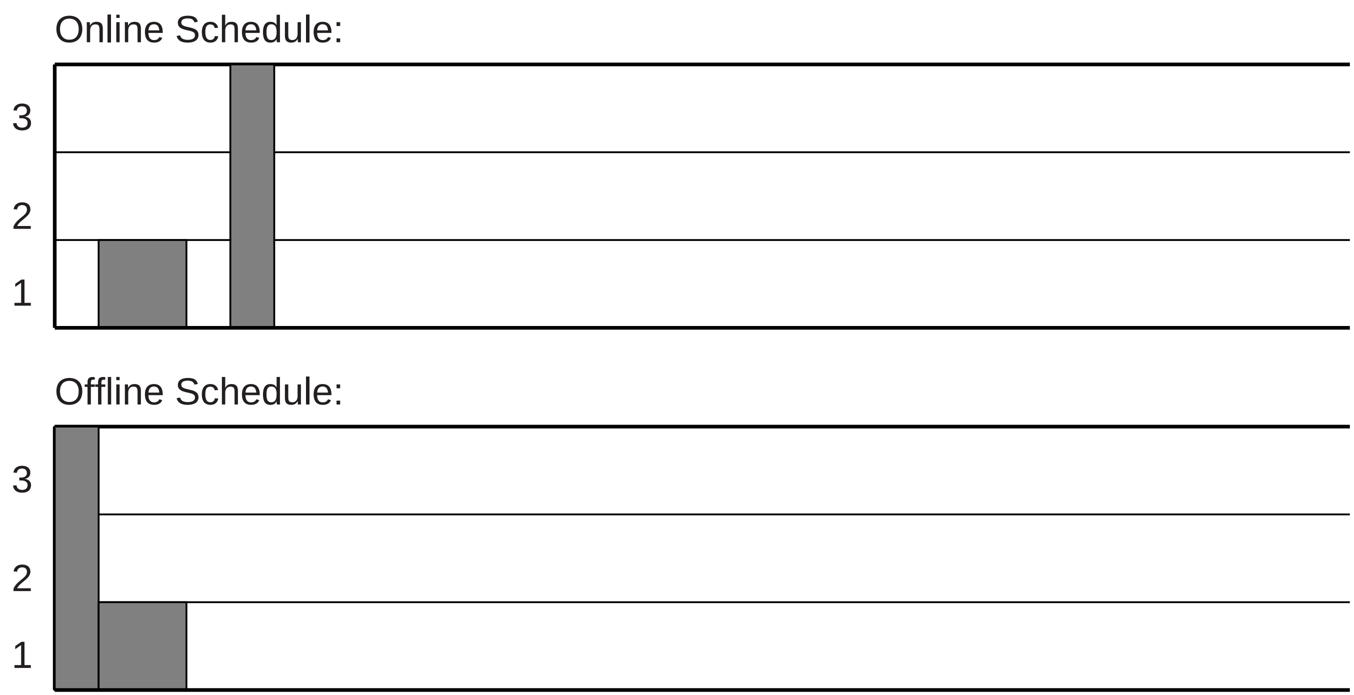
- For $m \geq 3$ the competitive ratio is at least $2$.



# *Case:* $m = 3$

Lower bound:

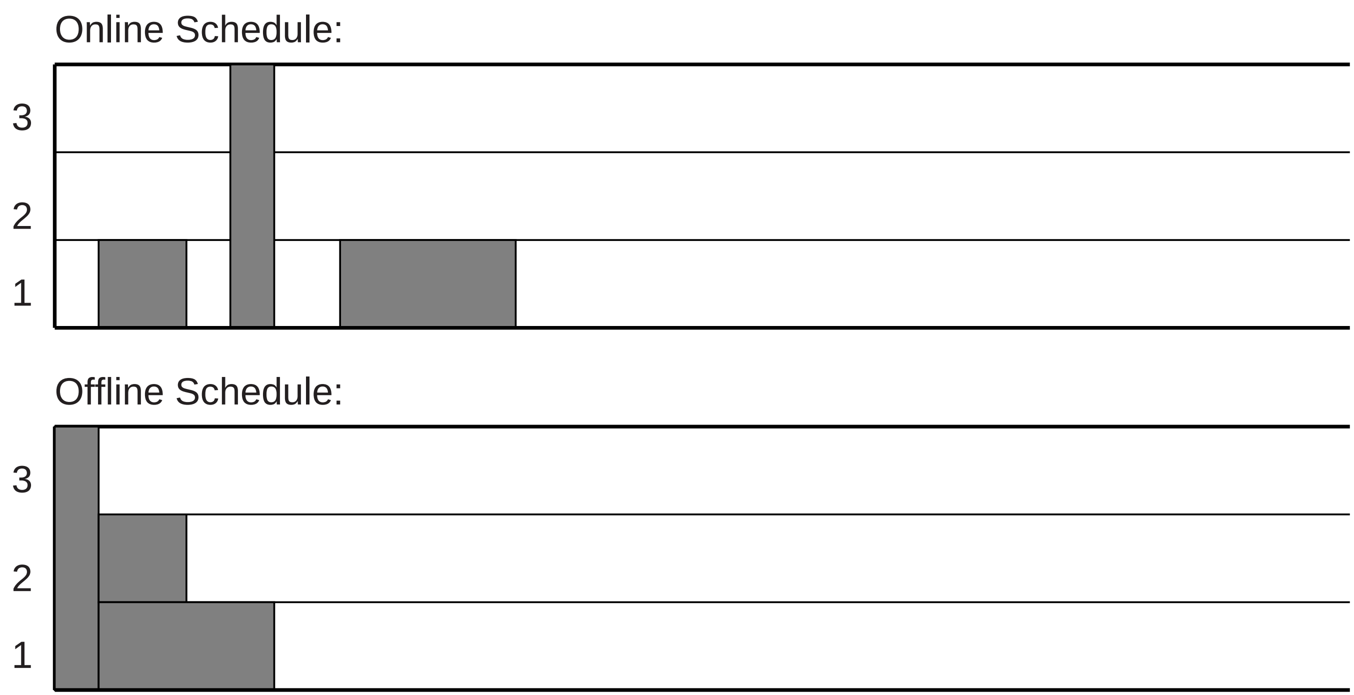
- For $m \geq 3$ the competitive ratio is at least $2$.



# *Case:* $m = 3$

Lower bound:

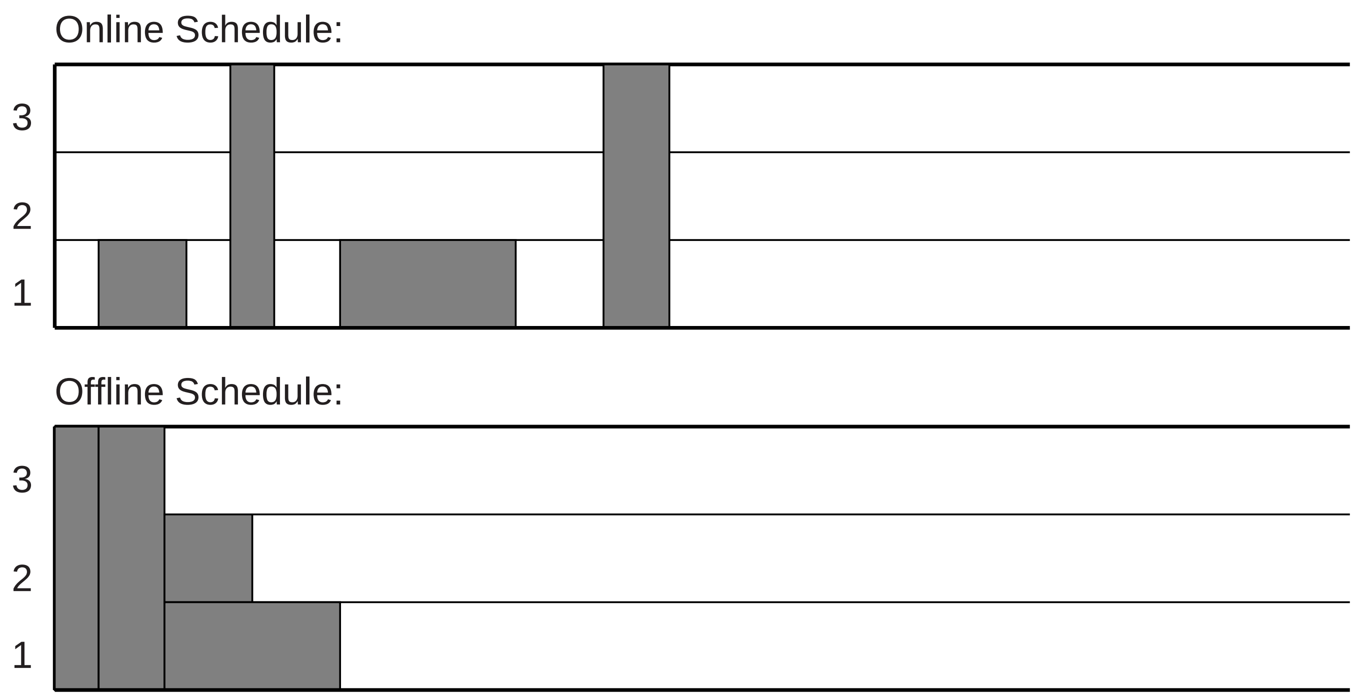
- For $m \geq 3$ the competitive ratio is at least $2$.



# *Case:* $m = 3$

Lower bound:

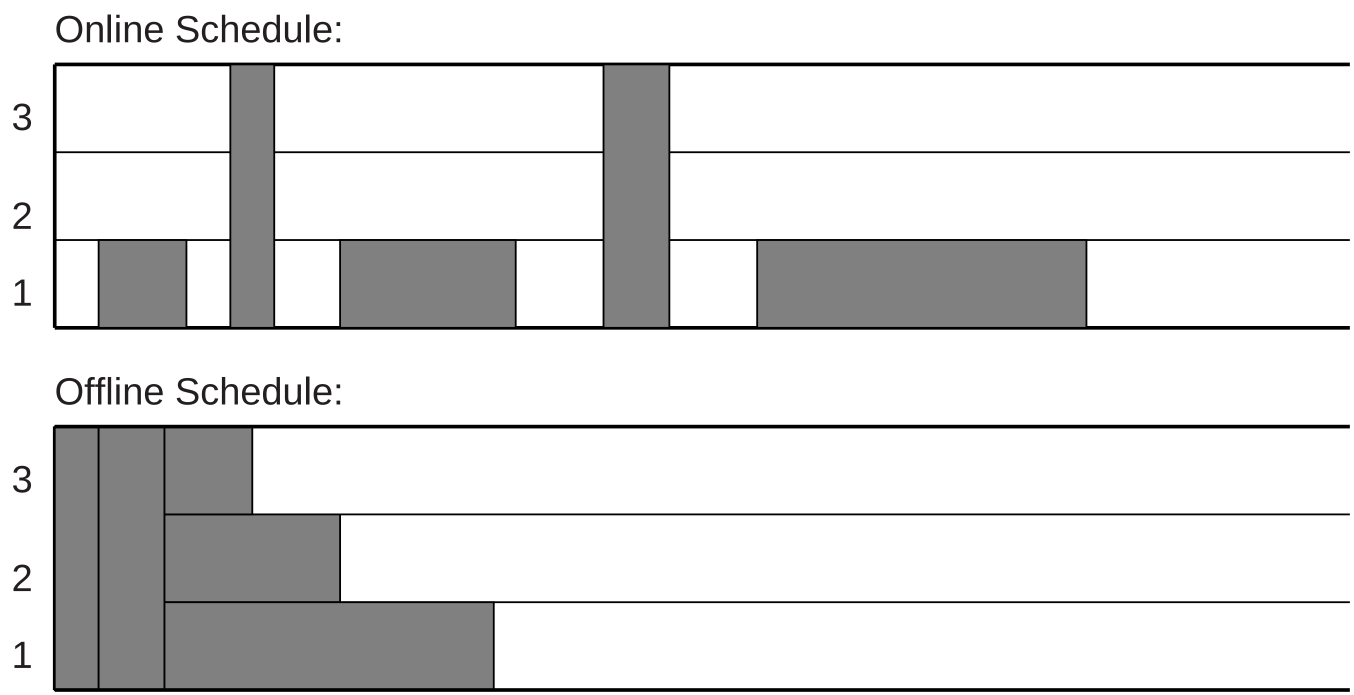
- For $m \geq 3$ the competitive ratio is at least $2$.



# *Case:* $m = 3$

Lower bound:

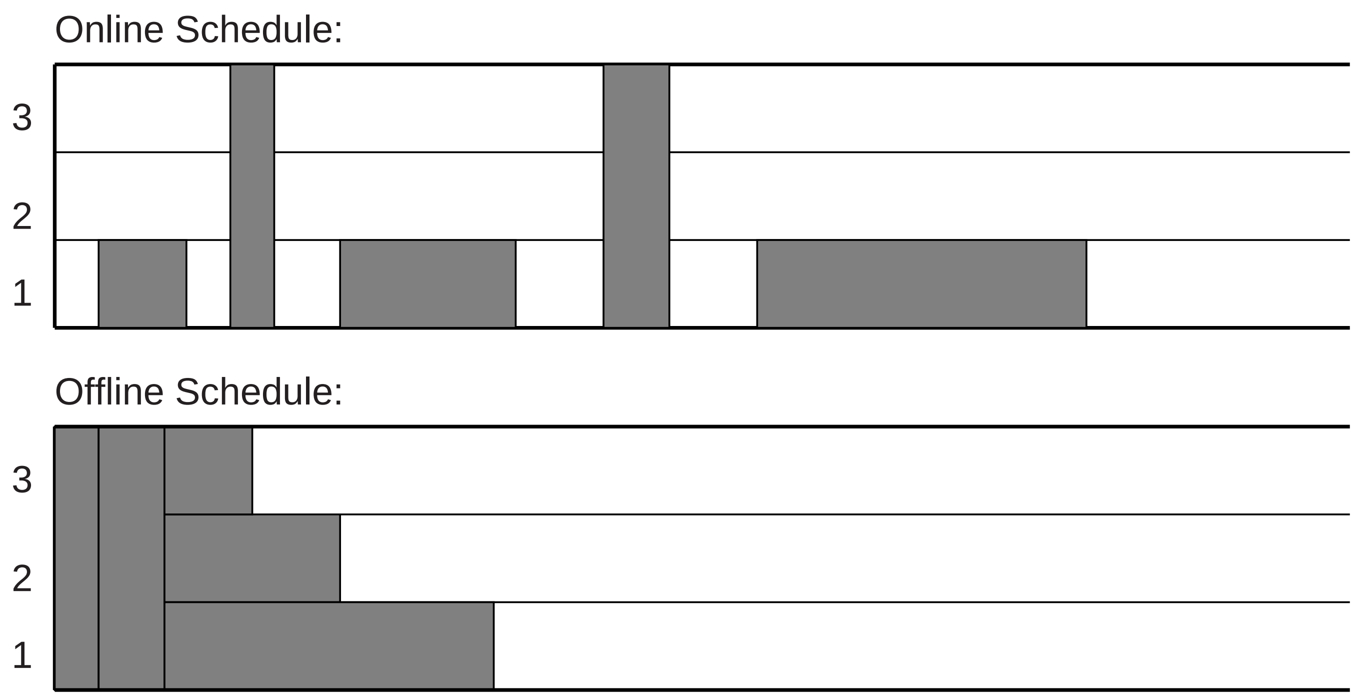
- For $m \geq 3$ the competitive ratio is at least $2$.



# *Case:* $m = 3$

Lower bound:

- For $m \geq 3$ the competitive ratio is at least $2$.



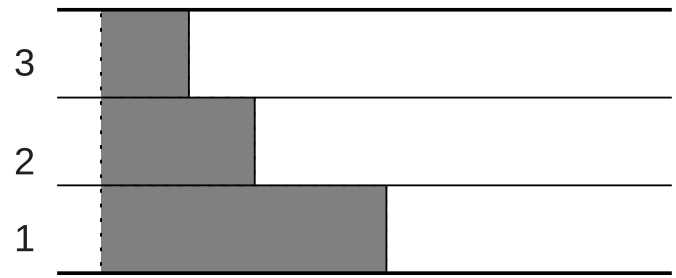

Upper bound:

- Greedy is $3$-competitive

# *Case:* $m = 3$

Algorithm $3M$:

- $m_j = 1$ job arrives:



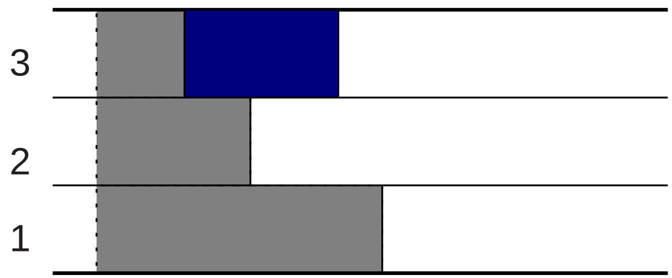

- $m_j = 2$ job arrives:

- $m_j = 3$ job arrives:

# *Case:* $m = 3$

Algorithm $3M$:

- $m_j = 1$ job arrives:




Decreasing profile is maintained. **Schedule Greedy.**

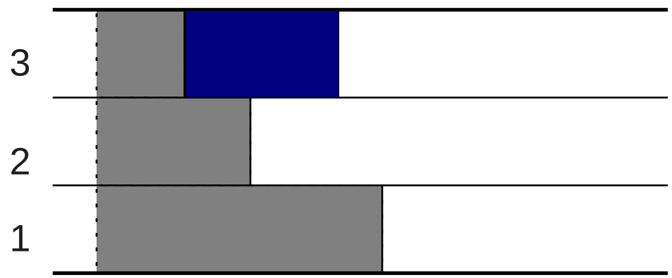
- $m_j = 2$ job arrives:

- $m_j = 3$ job arrives:

# *Case:* $m = 3$

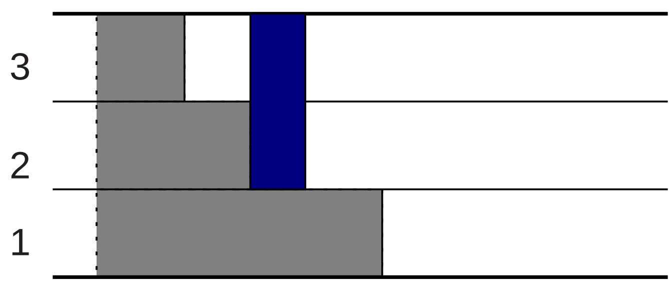
Algorithm $3M$:

- $m_j = 1$ job arrives:




Decreasing profile is maintained. **Schedule Greedy.**

- $m_j = 2$ job arrives:



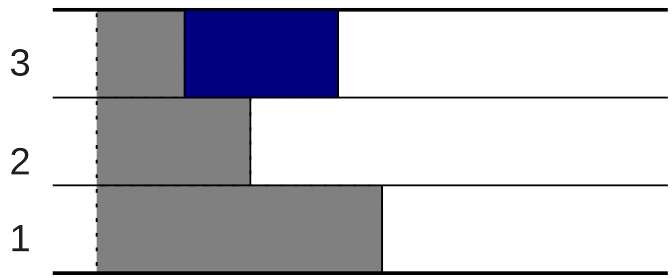

Not much is lost. **Schedule Greedy.**

- $m_j = 3$ job arrives:

# *Case:* $m = 3$

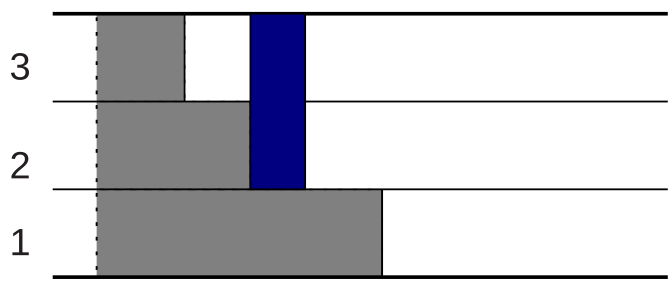
Algorithm $3M$:

- $m_j = 1$ job arrives:




Decreasing profile is maintained. **Schedule Greedy.**

- $m_j = 2$ job arrives:



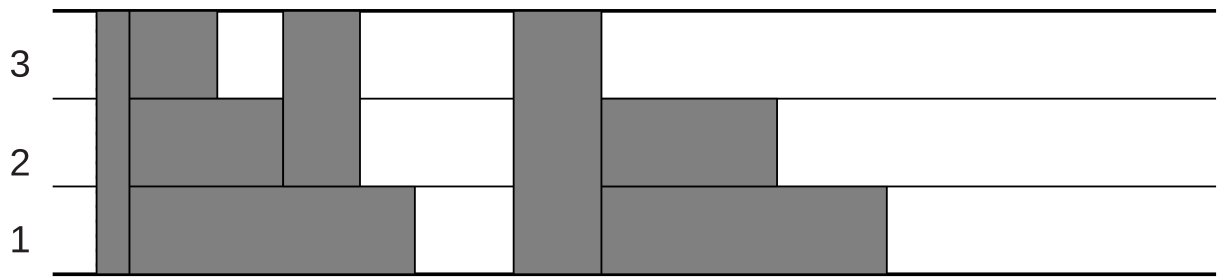

Not much is lost. **Schedule Greedy.**

- $m_j = 3$ job arrives: **Delay the job.**

# *Case:* $m = 3$

Algorithm $3M$:

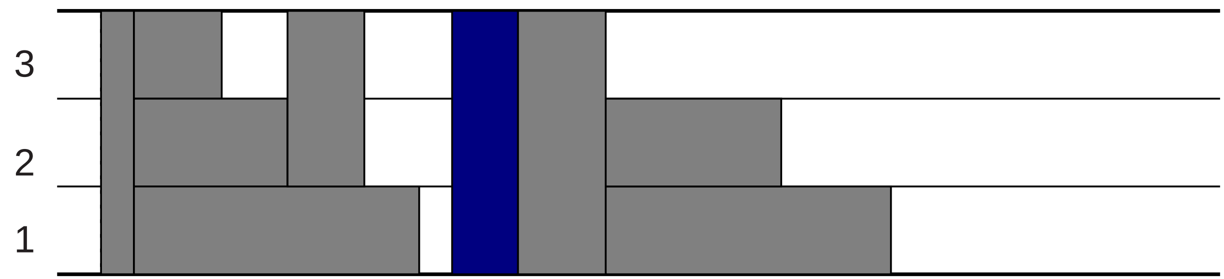
- $m_j = 3$ job arrives: **Delay the job.**



# *Case:* $m = 3$

Algorithm $3M$:

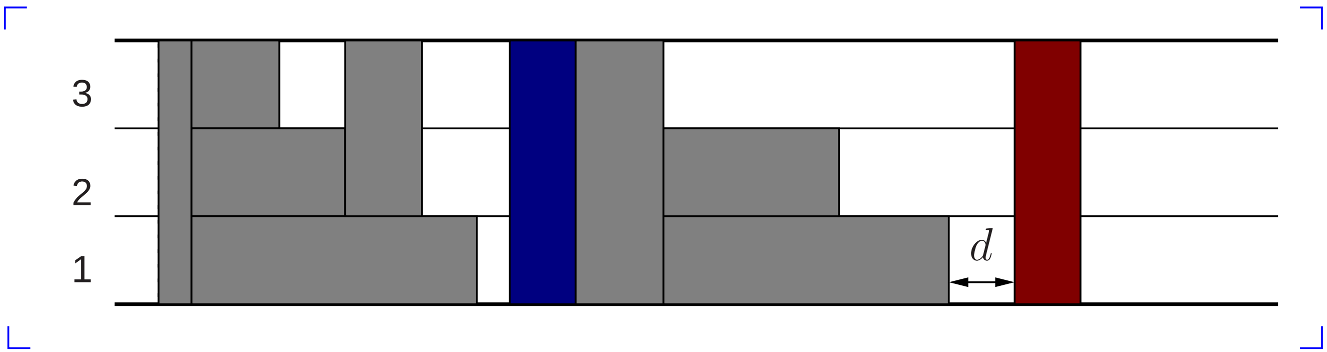
- $m_j = 3$ job arrives: **Delay the job.**



# *Case:* $m = 3$

Algorithm $3M$:

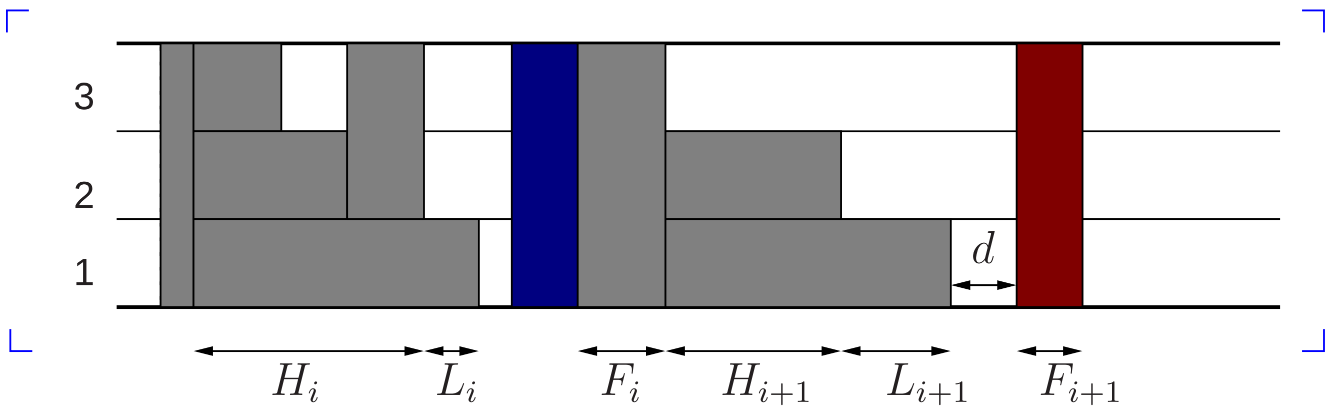
- $m_j = 3$ job arrives: **Delay the job.**



# *Case:* $m = 3$

Algorithm $3M$:

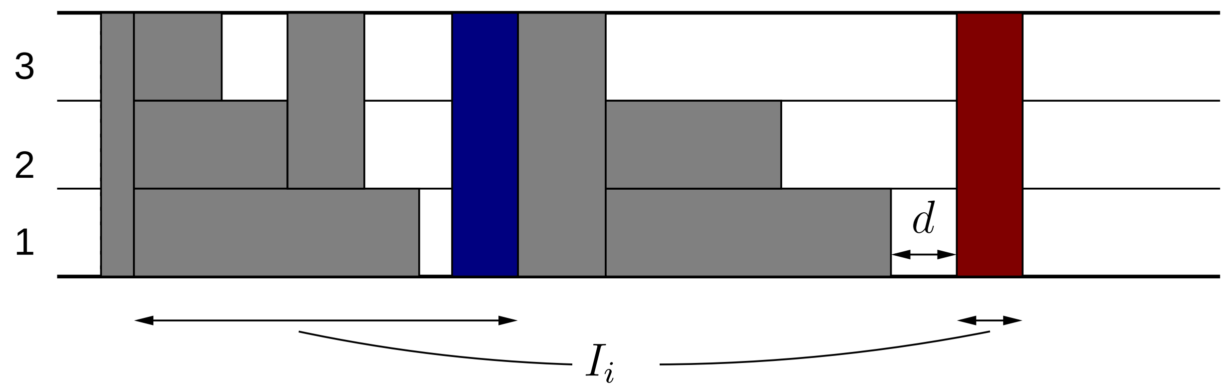
- $m_j = 3$ job arrives: **Delay the job.**




Define

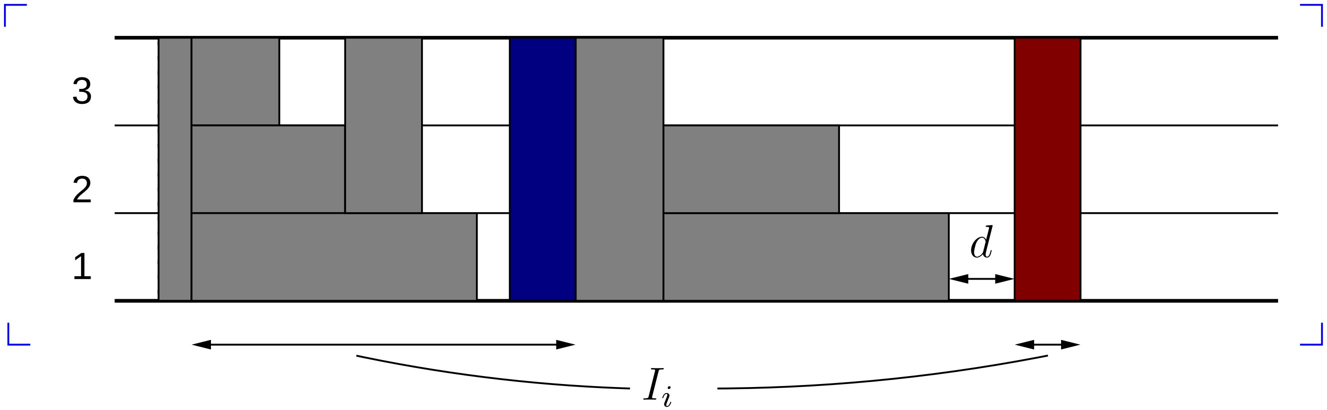
$$d := \left( \frac{1}{2} L_{i+1} - \frac{1}{4} H_{i+1} \right)^{+}$$

# *Case:* $m = 3$




$$C_{3M} = \sum I_i + (\text{last part of the schedule})$$

# *Case:* $m = 3$




$$C_{3M} = \sum I_i + (\text{last part of the schedule})$$

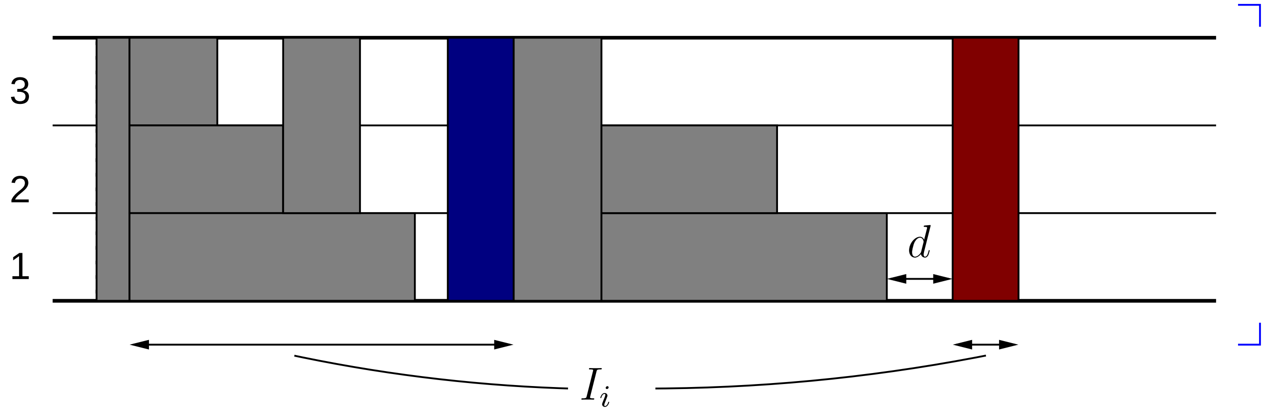
Improving the *load bound*:

$$\int_{I_i} load(t)\ dt > \frac{5}{3} I_i - F_{i+1}$$

Improving the *length bound*:

$$C^*(\sigma) \geq \sum F_i + \max_{j|m_j \leq 2} p_j$$

## *Case:* $m = 3$




$$C_{3M} = \sum I_i + (\text{last part of the schedule})$$

Improving the *load bound*:

$$\int_{I_i} load(t)\, dt > \frac{5}{3} I_i - F_{i+1}$$

Improving the *length bound*:

$$C^*(\sigma) \geq \sum F_i + \max_{j | m_j \leq 2} p_j$$

**Theorem 1** *Algorithm $3M$ is $2.8$-competitive.*

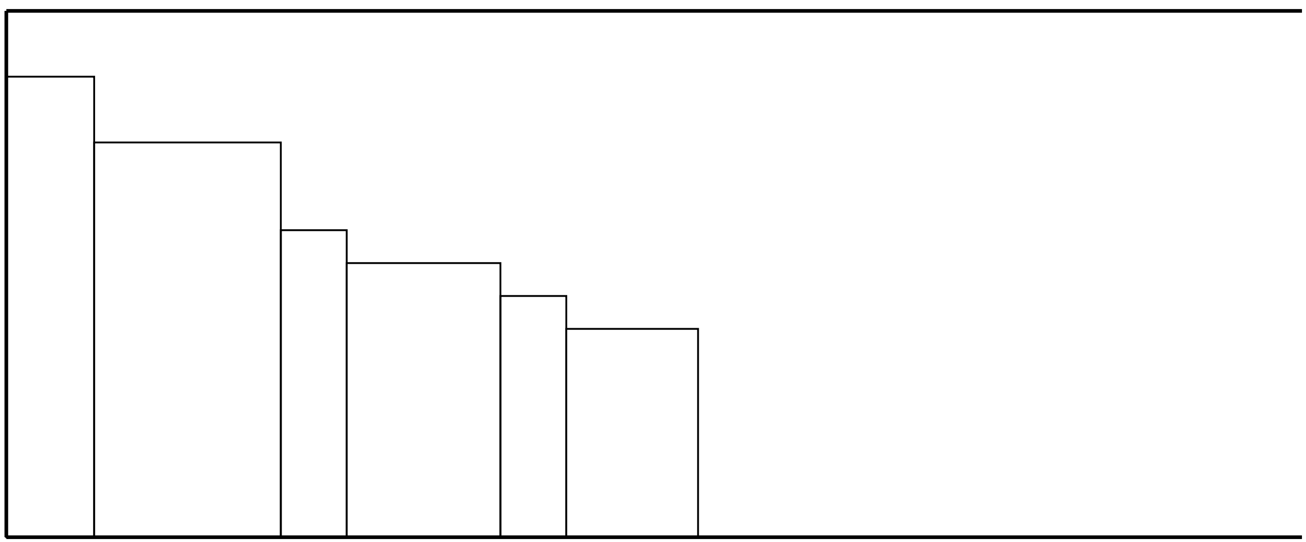
*Idea:* Either the *load bound* or the *length bound* works well. $\square$

# *Case: non-increasing $m_j$*

Greedy is $2.75$-competitive.

Algorithm **Modified Greedy** (MG):

- Schedule the jobs with $m_j > \frac{m}{3}$ one after the other.
- Schedule the other jobs greedily.

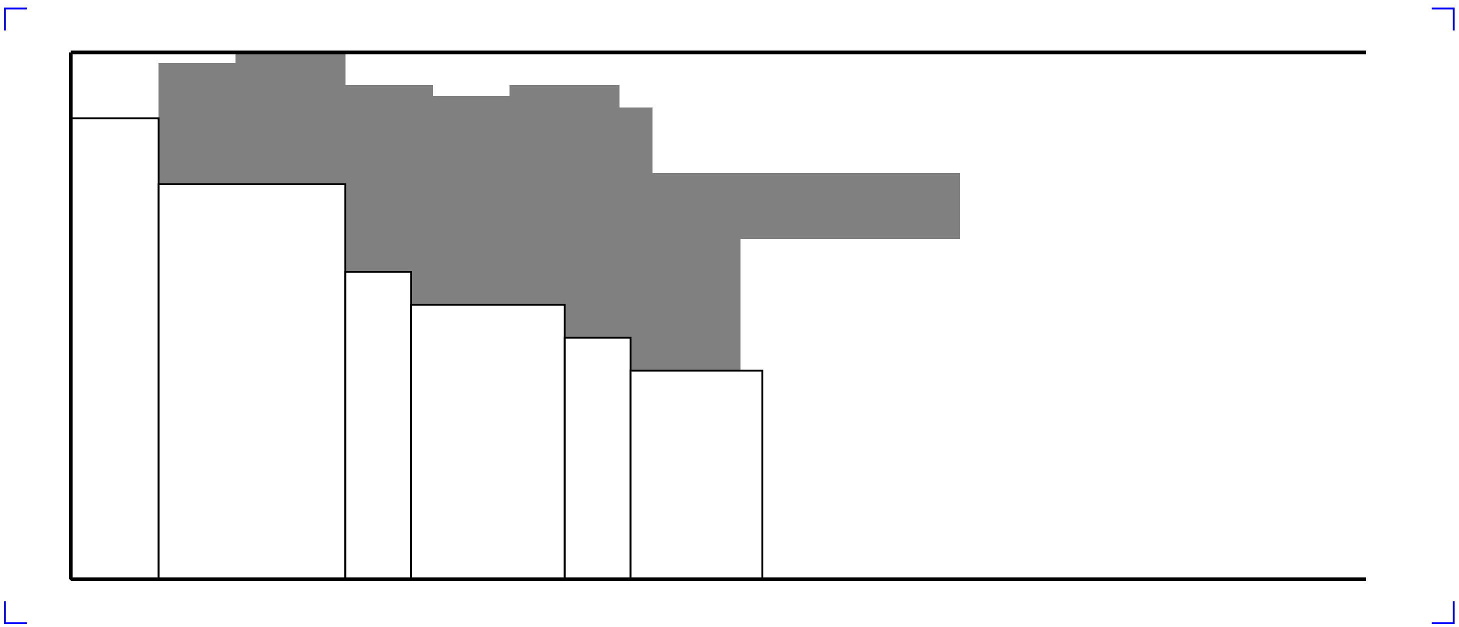


# *Case: non-increasing $m_j$*

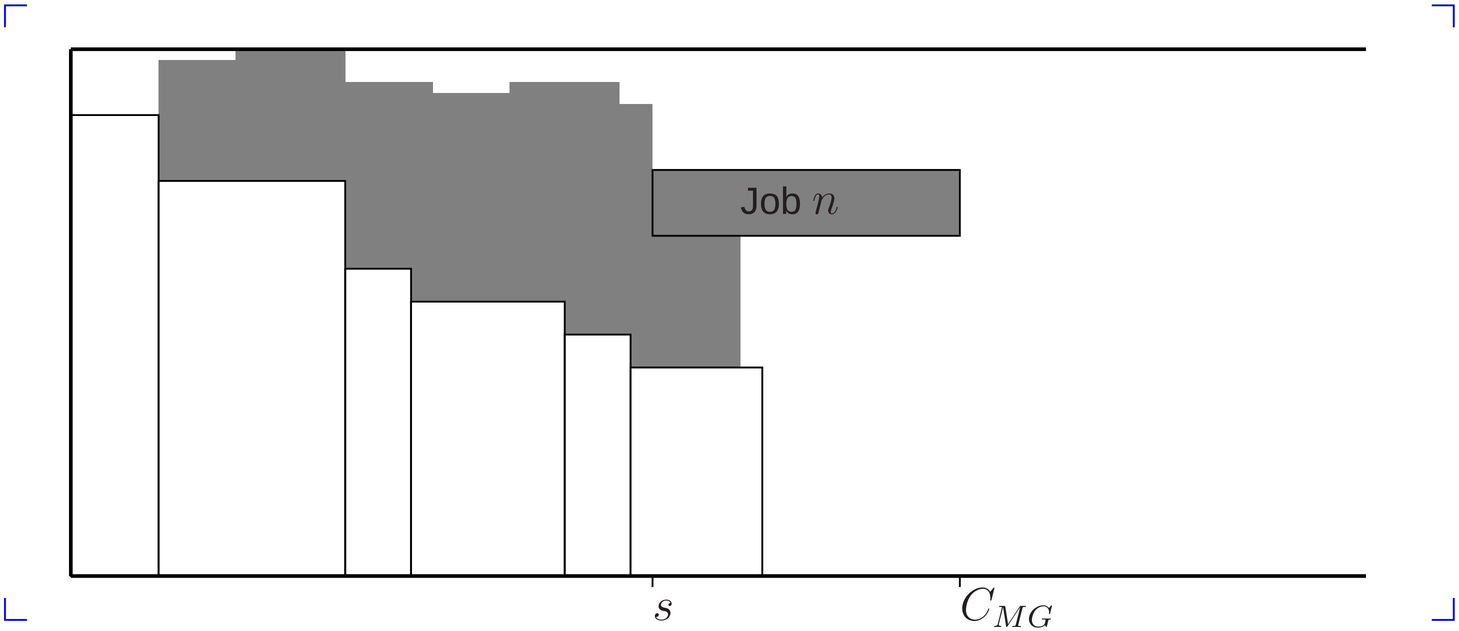
Greedy is $2.75$-competitive.

Algorithm **Modified Greedy** (MG):

- Schedule the jobs with $m_j > \frac{m}{3}$ one after the other.
- Schedule the other jobs greedily.



# *Case: non-increasing $m_j$*

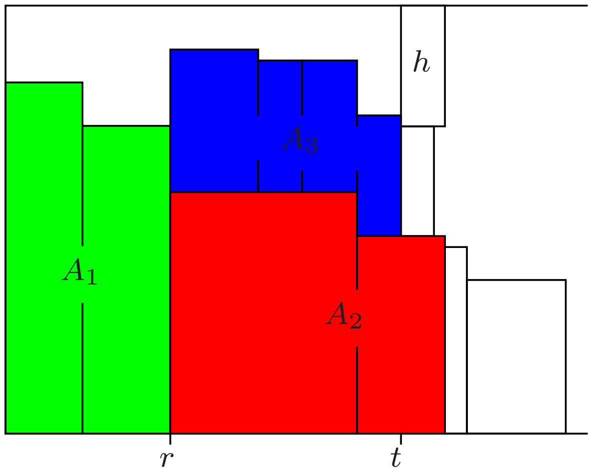



**Theorem 2** *Algorithm $MG$ is $2.5$-competitive.*

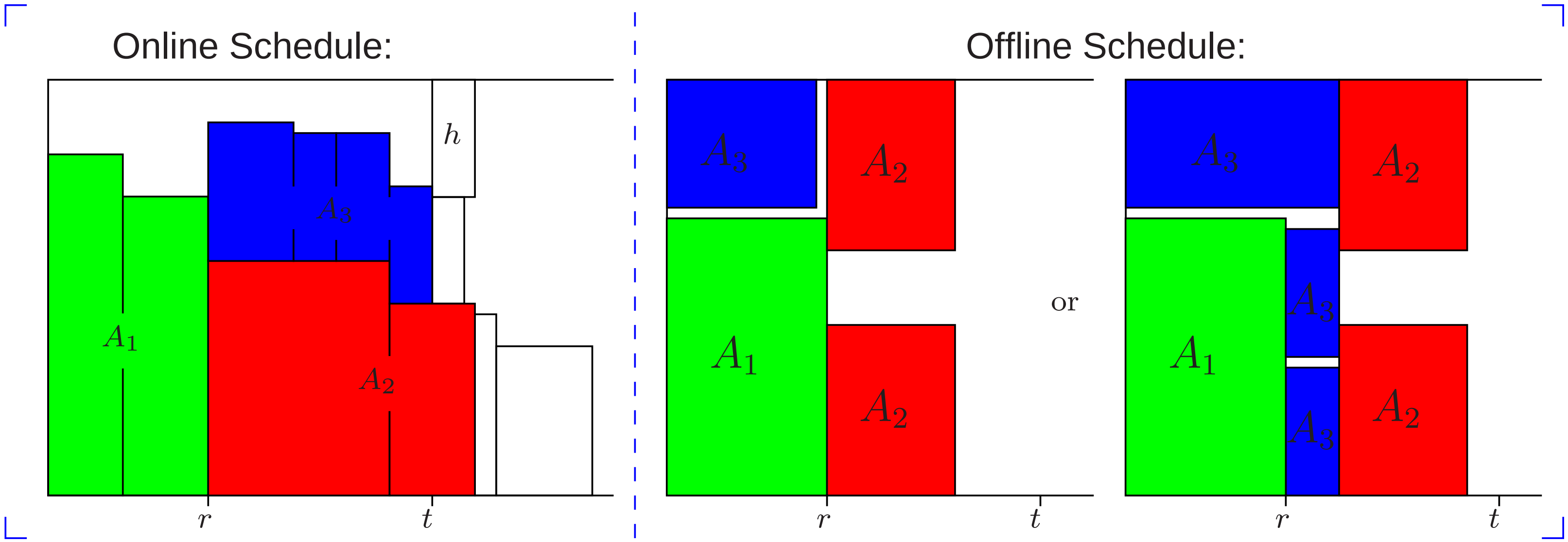
$$\begin{aligned} C_{MG}(\sigma) &= s + p_n \leq \frac{3}{2m} \sum_{i=1}^{n-1} m_i p_i + p_n \\ &= \frac{3}{2m} \sum_{i=1}^{n} m_i p_i + \left(1 - \frac{3m_n}{2m}\right) p_n \\ &\leq \frac{3}{2} C^*(\sigma) + \left(1 - \frac{3m_n}{2m}\right) C^*(\sigma) \leq \frac{5}{2} C^*(\sigma) \end{aligned}$$

# *Case: non-increasing* $m_j$

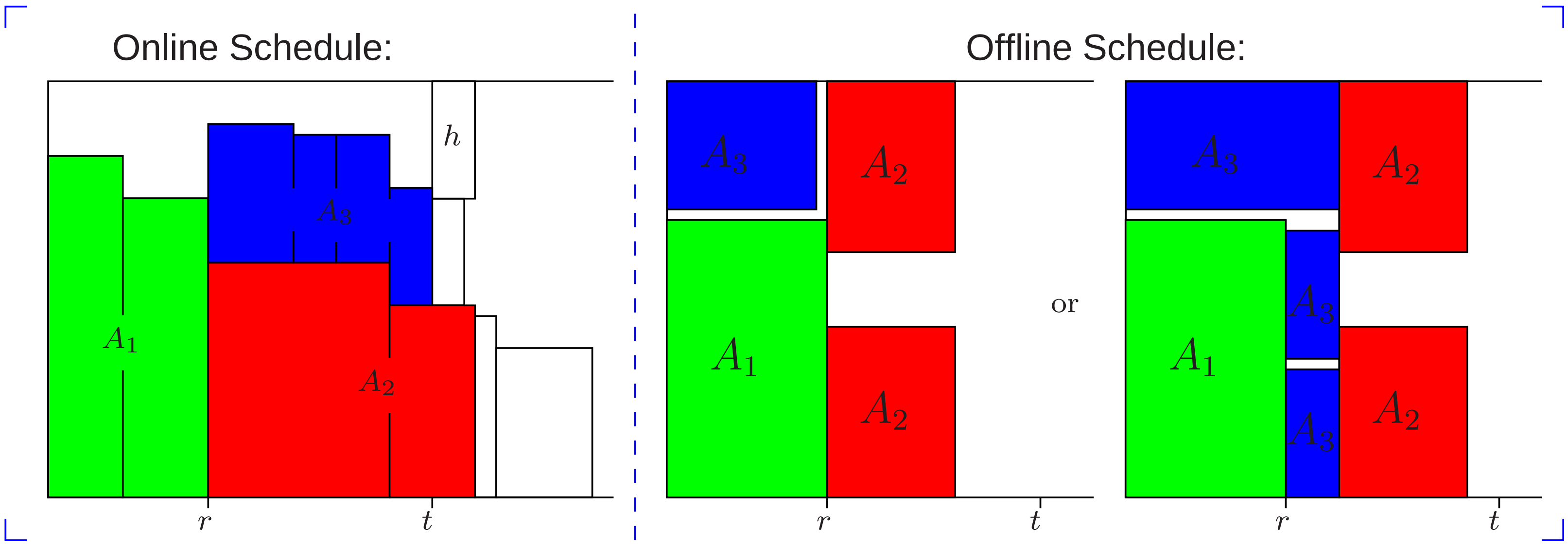
Online Schedule:



# *Case: non-increasing* $m_j$

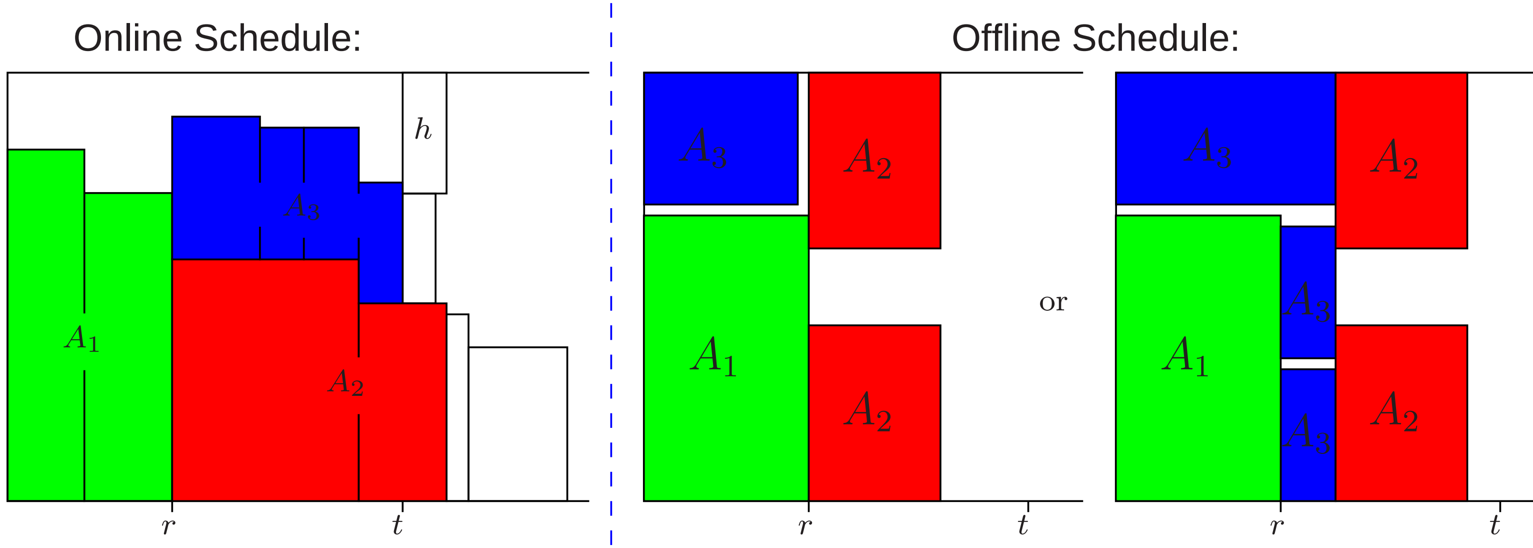


# *Case: non-increasing* $m_j$




$$C^* = |A_1| + \frac{1}{2}|A_2|$$
$$\geq r + \frac{1}{2}(t-r) = \frac{1}{2}(r+t) \quad \geq_{(r \geq \frac{1}{2}t)} \quad \frac{3}{4}t$$

# *Case: non-increasing* $m_j$




$$
\begin{aligned}
C^* &= |A_1| + \frac{1}{2}|A_2| \\
&\geq r + \frac{1}{2}(t-r) = \frac{1}{2}(r+t) \qquad \geq_{(r \geq \frac{1}{2}t)} \quad \frac{3}{4}t
\end{aligned}
$$

$$
\begin{aligned}
C^* &= |A_1| + \frac{1}{2}|A_2| + \frac{1}{3}(|A_3| - |A_1|) \\
&\geq r + \frac{1}{2}(t-r) + \frac{1}{3}(t-2r) = \frac{5}{6}t - \frac{1}{6}r \qquad \geq_{(r \leq \frac{1}{2}t)} \quad \frac{3}{4}t
\end{aligned}
$$

# *Case: non-increasing* $m_j$

New bound on $C^*$:

$$C^* \geq \frac{3}{4}t$$

**Theorem 3** *Algorithm $MG$ is $\frac{67}{27}$-competitive ($\approx 2.4815$).*

*Idea:* Case distinction on $t/C_{MG}$. $\square$

## *Questions?*

Case: $m = 3$:

$$2 \leq \textsf{?} \leq 2.8$$

Case: non-increasing $m_j$

$$1.88 \leq \textsf{?} \leq 2.4815$$