Improved Online Scheduling in Maximizing Throughput of Equal Length Jobs

Nguyen Kim Thang
(university Paris-Dauphine, France)

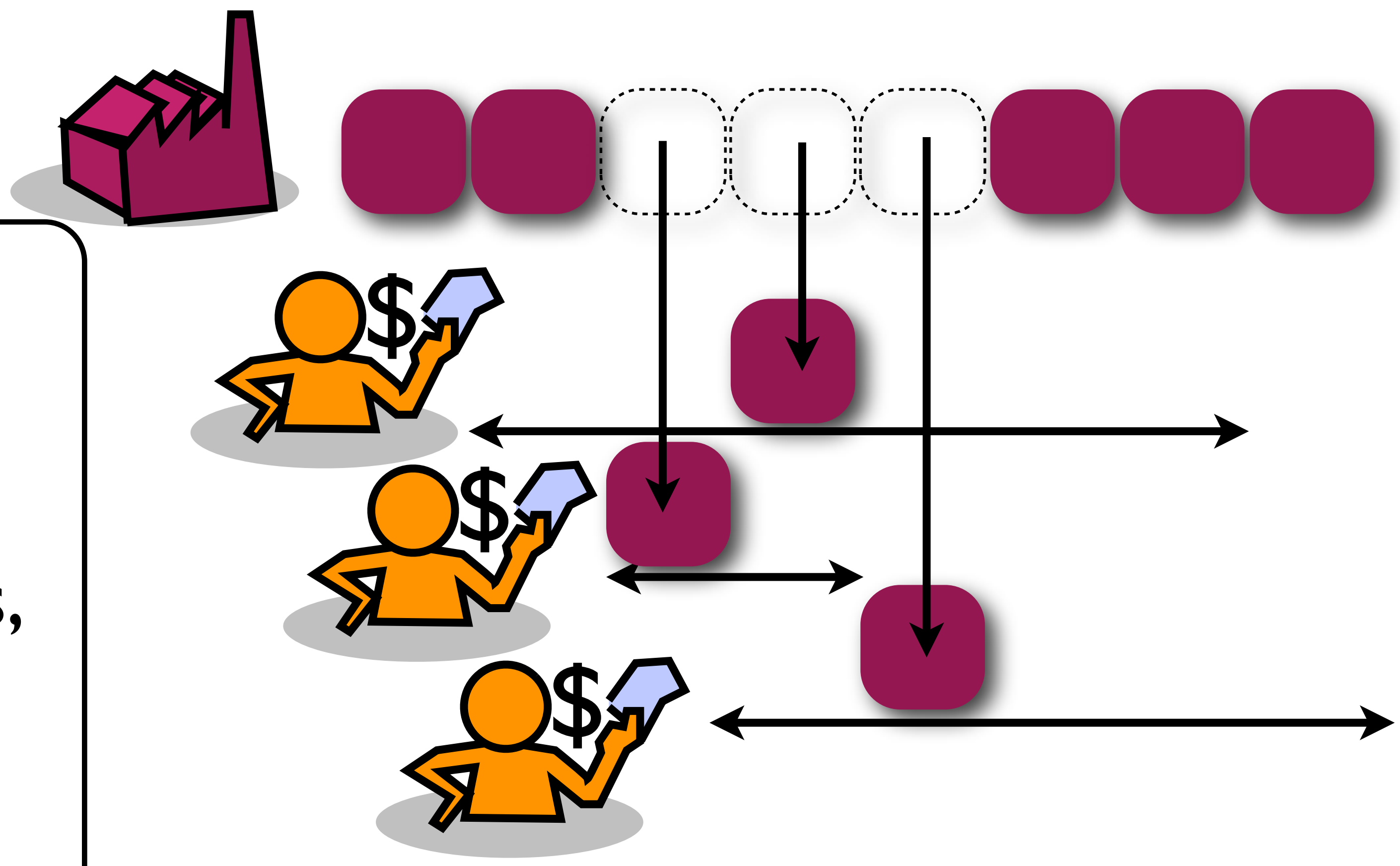
Motivation

Profit maximization

Enterprise: perishable product (electricity, ice-cream, ...).

Clients: single-minded, arrive online, different demands.

Goal: maximize the profit.



Other applications

Data broadcast: online broadcast pages

ATM network: online packages, typically of the same length.

Objective: maximize the total value.

Model

Online Scheduling

Jobs: arrive at r_i, processing time p_i, deadline d_i, value (weight) w_i.

Objective: maximize the total value of jobs completed on time.

Preemption is necessary

- Preemption with restart: when a job is scheduled again, it must be executed from the beginning (e.g., data broadcast).
- Preemption with resume: when a job is scheduled again, the previously done work can be resumed (e.g., ATM network) .

Competitive ratio

- An algorithm ALG is α-competitive if for any instance I

$$\frac{OPT(I)}{ALG(I)} \leq \alpha \quad \text{(maximization problem)}$$

Competitive ratio

- An algorithm ALG is **α-competitive** if for any instance I

$$\frac{OPT(I)}{ALG(I)} \leq \alpha \quad \text{(maximization problem)}$$

- What is a competitive ratio?
  - Measure the performance of an algorithm (worst-case analysis)
  - The price of an object (the problem):

Algorithm (upper bound) ← negotiation → Adversary (lower bound)

Contribution

	equal processing times	bounded processing times (by k)	unbounded processing times
unit weight	$\alpha = 1$	$\alpha = \Theta(\log k)$	∞
general	$\frac{3\sqrt{3}}{2} \leq \alpha \leq 5$ ≤ 4.24	$\alpha = \Theta(k/\log k)$	∞

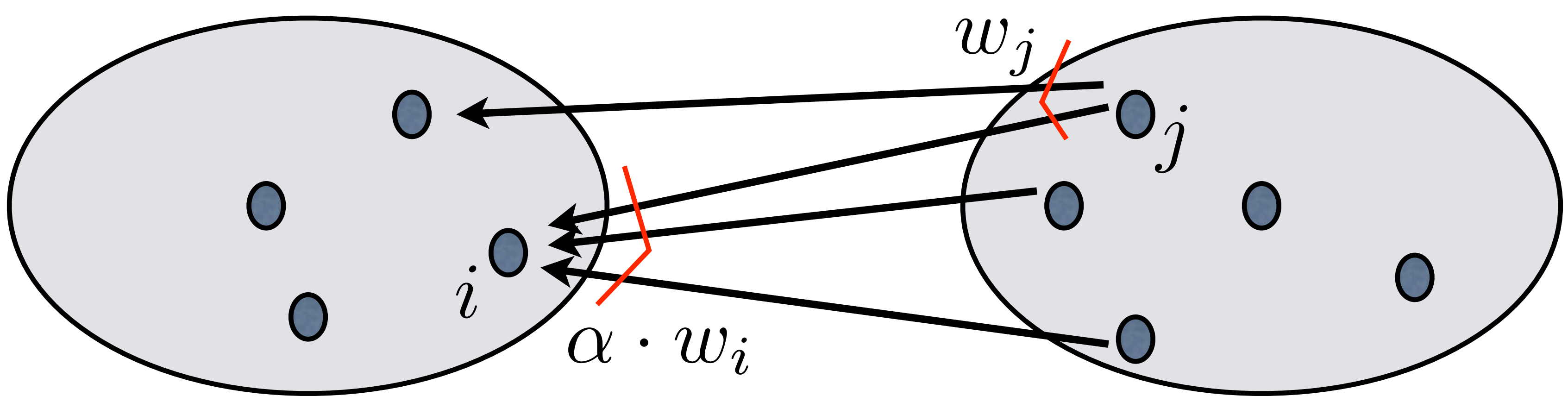
Contribution

	equal processing times	bounded processing times (by k)	unbounded processing times
unit weight	$\alpha = 1$	$\alpha = \Theta(\log k)$	∞
general	$\frac{3\sqrt{3}}{2} \leq \alpha \leq 5$ ≤ 4.24	$\alpha = \Theta(k/\log k)$	∞

- ☑ **Improved algorithms for both models of preemption**
- ☑ **Weights and correlation between jobs' deadlines**

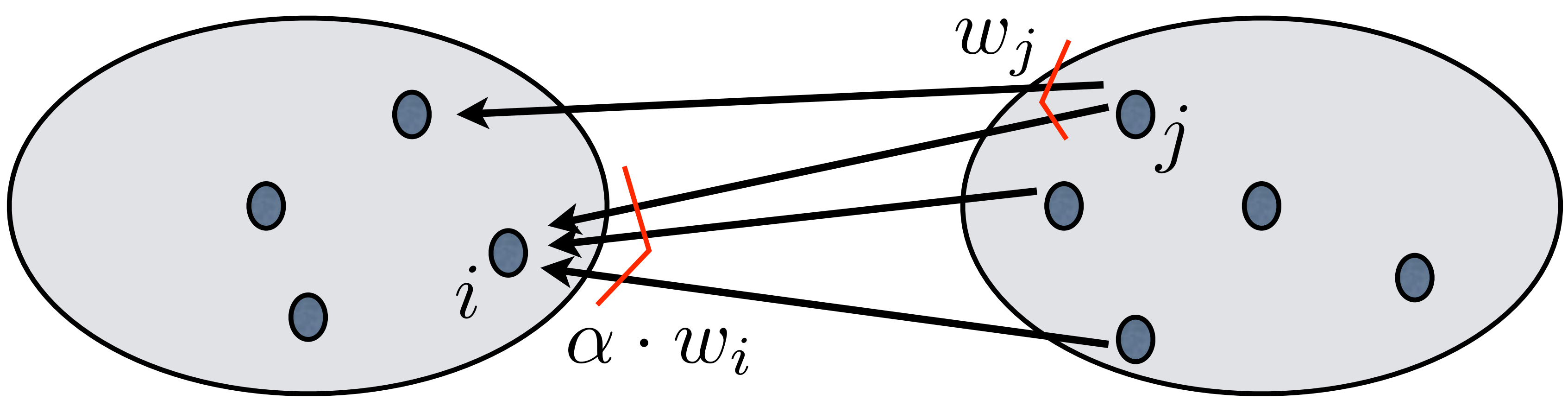
Settling the competitivity

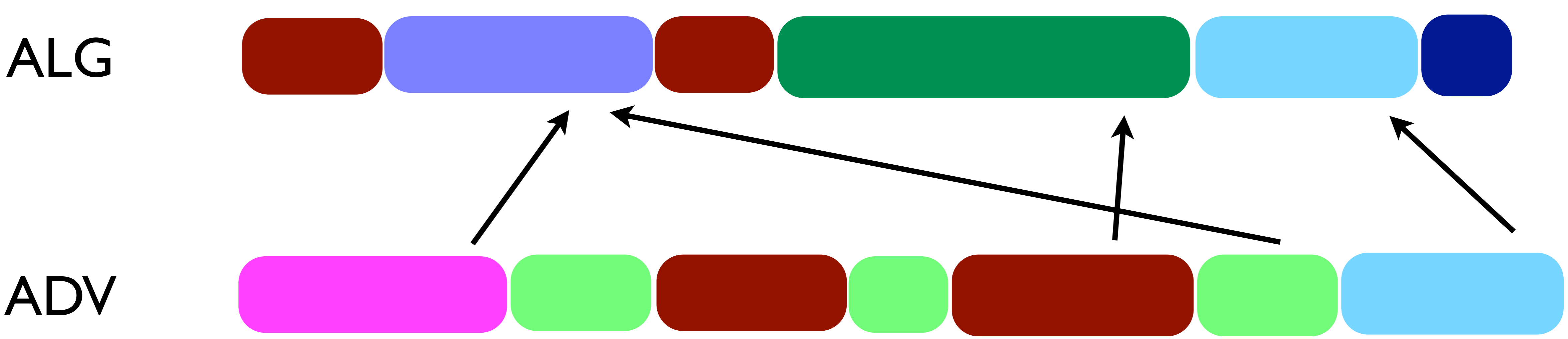
- Methods: charging scheme, potential function, etc



Settling the competitivity

- Methods: charging scheme, potential function, etc





Starting point

- Paradox: low weight, imminent deadline ⟷ (which jobs?) higher weight, later deadline

Starting point

- Paradox: low weight, imminent deadline ⟷ (which jobs?) higher weight, later deadline
- p : initial job length, $q_j(t)$: length of job j at time t
- A job j is pending at time t if $t + q_j(t) \leq d_j$

Starting point

- Paradox: low weight, imminent deadline ⟷ (which jobs?) higher weight, later deadline
- p : initial job length, $q_j(t)$: length of job j at time t
- A job j is pending at time t if $t + q_j(t) \leq d_j$

- A 5-competitive algorithm (preemption with restart)

  At any time

  - If no currently scheduled job, schedule the pending one with highest weight
  - If a new pending job arrive with weight at least twice that of the currently scheduled job, then schedule the new one (by interrupting the current job)

Observations

- Correlation among jobs' deadlines is ignored

- Treatment:
  - A job i is urgent at time t if $d_i < t + q_i(t) + p$
  - Some job would be delayed by new urgent jobs (even with low weight)
  - Ensure no significant lost if new heavy jobs arrive.

Algorithm

- Initially, set $Q = \emptyset, \alpha = 0, 1 < \beta < 3/2$
- At time t, let i, j be a new released job and the currently scheduled job, respectively. At any interruption, if $\alpha > 0$ then $\alpha := \alpha + 1$

Algorithm

- Initially, set $Q = \emptyset, \alpha = 0, 1 < \beta < 3/2$

- At time t, let i, j be a new released job and the currently scheduled job, respectively. At any interruption, if $\alpha > 0$ then $\alpha := \alpha + 1$

  - **If** $w_i \geq 2w_j, w_i \geq 2^\alpha w(Q)$ **do**
    - schedule i
    - set $Q = \emptyset, \alpha = 0$

Algorithm

- Initially, set $Q = \emptyset, \alpha = 0, 1 < \beta < 3/2$

- At time t, let i, j be a new released job and the currently scheduled job, respectively. At any interruption, if $\alpha > 0$ then $\alpha := \alpha + 1$

  - **If** $w_i \geq 2w_j, w_i \geq 2^\alpha w(Q)$ **do** schedule i, set $Q = \emptyset, \alpha = 0$

  - **If** $\alpha = 0, \beta w_j \leq w_i \leq 2w_j$, i urgent and $d_j \geq t + 2p$ **do** schedule job which is $\arg\max\{w_\ell : d_\ell < t + 2p\}$, set $Q = \{j\}, \alpha = 1$

Algorithm

- Initially, set $Q = \emptyset, \alpha = 0, 1 < \beta < 3/2$

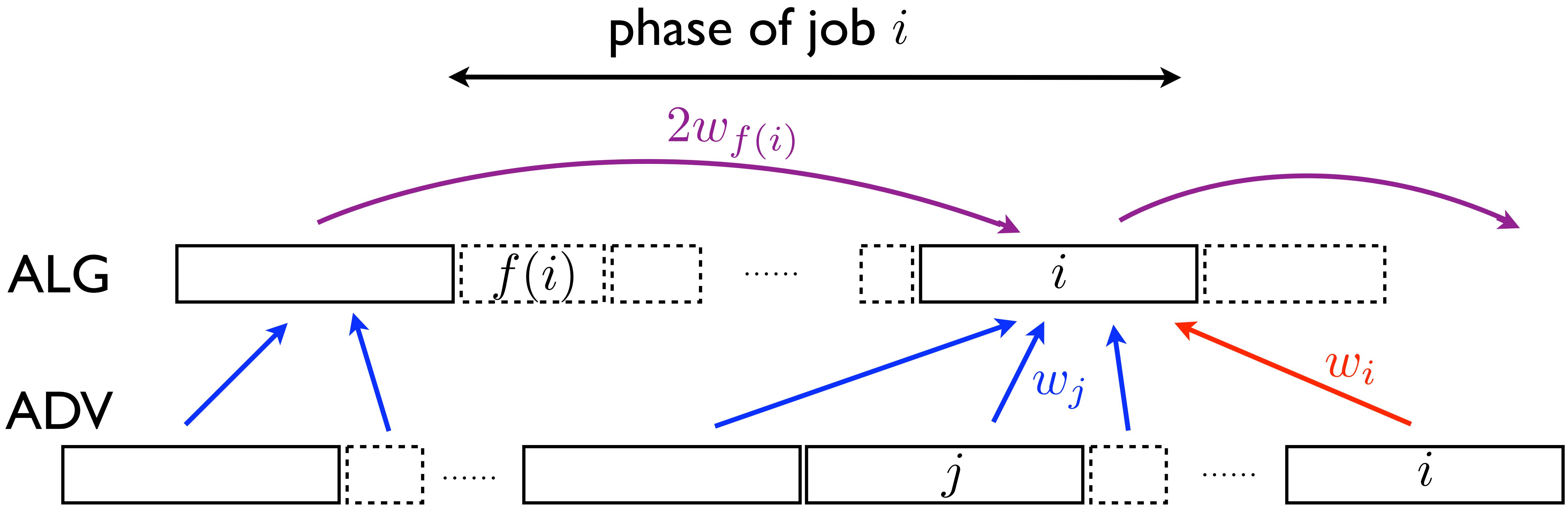
- At time t, let i, j be a new released job and the currently scheduled job, respectively. At any interruption, if $\alpha > 0$ then $\alpha := \alpha + 1$

  - **If** $w_i \geq 2w_j, w_i \geq 2^\alpha w(Q)$ **do** schedule i; set $Q = \emptyset, \alpha = 0$

  - **If** $\alpha = 0, \beta w_j \leq w_i \leq 2w_j$, i urgent and $d_j \geq t + 2p$ **do** schedule job which is $\arg\max\{w_\ell : d_\ell < t + 2p\}$; set $Q = \{j\}, \alpha = 1$

  - **If** i is urgent, $w_i \geq 2w_j + w_{j'}$, no job ℓ such that $S_j(t) + 2p \leq d_\ell < t + 2p, w_\ell \geq w_j$ **do** schedule i

The charging scheme



- Theorem: the algorithm is $(2+\sqrt{5})$-competitive

- Theorem: there is a $(2+\sqrt{5})$-competitive algorithm for model of preemption with resume

Conclusion

- ☑ Improved algorithms for both models of preemption

- ☐ Open questions:
  - Settling the right competitive ratio $2.5 \leq \alpha \leq 4.24$
  - Interesting: not to reduce the gap but new methods.

Thank you!

Thank Kristoffer!