CS6382 Algorithm Analysis and Game Theory

Game Theory in Scheduling

Classic Scheduling Problems

- Set of m parallel machines M
- Set of n jobs J: each job j has (in most general case)
  - Arrival time r_j
  - Processing time (size) p_{ij} on machine $i \in M$
  - Value v_j
  - Deadline d_j

Classic Scheduling Problems

- A job j is considered **completed** if there exists machine i such that job j is processed for p_{ij} time units on i during $[a_j, d_j]$.
  - Let C_j denote the completion time of job j.
  - $C_j = \infty$ if job j is not completed.

- Preemptive vs. Non-preemptive
  - Whether job j is processed **continuously** on machine i

Objective of the Scheduler

- Schedule all jobs:
  - **Minimize the makespan $\max_{j \in J} C_j$**
  - Minimize total completion time $\sum_{j \in J} C_j$
  - Minimize total flow time $\sum_j (C_j - a_j)$

- Schedule some jobs to **maximize the total value $\sum_{j: C_j \neq \infty} v_j$**

Objective of the Machines

- Minimize the workload assigned to it.

Objective of the Machines

- Minimize the workload assigned to it.

- If there are payments for working:
  - maximize the profit: payment – workload

Objective of the Machines

- Minimize the workload assigned to it.

- If there are payments for working:
  - maximize the profit: payment – workload

- Suppose the machines are controlled by **selfish agents**:
  - Only machine i (agent i) knows p_{ij}, for all jobs $j \in J$
  - Machines may lie about p_{ij} to maximize their own utilities.

Objective of the Jobs

- Minimize its own completion time.
- Maximize its profit: v_j if completed; 0 otherwise.

Objective of the Jobs

- Minimize its own completion time.
- Maximize its profit: v_j if completed; 0 otherwise.

- If there is charging for being processed:
  - maximize the profit: value - charging

Objective of the Jobs

- Minimize its own completion time.
- Maximize its profit: v_j if completed; 0 otherwise.

- If there is charging for being processed:
  - maximize the profit: value - charging

- Suppose the jobs are controlled by **selfish agents**:
  - Only job j (agent j) knows a_j, v_j, d_j, p_{ij}
  - Jobs may lie about its information to maximize its own utility.

Game theory

- Compute a **Nash Equilibrium**: no agent has incentive to unilaterally change strategy.
- Design a scheduling mechanism under which **selfish machines** are truthful: misreporting is not beneficial.
- Design a scheduling mechanism under which **selfish jobs** are truthful: misreporting is not beneficial.

Nash Equilibrium

Nash Equilibrium

- Price of Anarchy (POA) : ratio between worst possible Nash equilibrium and the social optimum
  - All information are public.
  - How much worse will it be, if agents behave selfishly?

- Congestion Model
- Sequencing Model
- ...

Congestion model

- Selfish job chooses machine
- Utility: minimize completion time on the chosen machine.
- Objective of scheduler: minimize the makespan

Classic Results on Congestion Model

- On identical machines
  - POA= $\frac{3}{2}$ for 2 machines [Koutsoupias & Papadimitriou, STACS 1999].
  - POA= $\Omega\left(\frac{\log m}{\log\log m}\right)$ for m machines [Koutsoupias & Papadimitriou, STACS 1999]
  - POA= $O\left(\frac{\log m}{\log\log m}\right)$ for m machines [Czumaj & Vocking, SODA 2002]
- On related machines:
  - POA $\geq \Phi \approx 1.618$ on 2 machines [Koutsoupias & Papadimitriou, STACS 1999].
  - POA= $\Theta(\frac{\log m}{\log\log\log m})$ for m machines [Czumaj & Vocking, SODA 2002]

Sequencing Model

- Utility of jobs: their own completion time
- Objective: minimize the makespan
- Need to define how jobs on the same machine are scheduled

Classic Results on Sequencing Model

- Shortest Processing Time first (SPT)
  - $2 - \frac{2}{m+1}$ for identical machines; $\Theta(\log m)$ on related machines; $O(m)$ for unrelated machines. [Immorlica et al., WINE 2005]
- Longest Processing Time first (LPT)
  - $\frac{4}{3} - \frac{1}{3m}$ for identical machines. [Christodoulou et al., ICALP 2004]
  - $\frac{4}{3} - \frac{1}{3m} \leq \mathrm{POA} \leq 2 - \frac{2}{m}$ for related machines. [Immorlica et al., WINE 2005]

Selfish Machines

Model

- There are n jobs to be completed ($a_j = 0$, no deadline or value).
- Every machine i claims a processing time t_{ij} for processing job j.

- Scheduler assigns jobs to machines, and determines a payment for every machine, with the objective of minimizing makespan.
  - The actual processing time of j on i is given by p_{ij}, which might be different from t_{ij}.

Model

- Let $x_{ij} \in \{0,1\}$ be indicator of job j assigned to machine i
- Let $f_i(x)$ be the total payment to machine i.

- Utility of machine i: $f_i(x) - \sum_{j:x_{ij}=1} p_{ij}$

- Makespan: $\max_i \left\{ \sum_{j:x_{ij}=1} p_{ij} \right\}$

Truthful Mechanism Design

- If there is no payment:
  - All machines claim $t_{ij} = \infty$ to get rid of working.

Truthful Mechanism Design

- If there is no payment:
  - All machines claim $t_{ij} = \infty$ to get rid of working.

- Is there any scheduling algorithm + payment function that results in a truthful mechanism?

Truthful Mechanism Design

- If there is no payment:
  - All machines claim $t_{ij} = \infty$ to get rid of working.

- Is there any scheduling algorithm + payment function that results in a truthful mechanism? **YES!**

Truthful Mechanism Design

- If there is no payment:
  - All machines claim $t_{ij} = \infty$ to get rid of working.

- Is there any scheduling algorithm + payment function that results in a truthful mechanism?

- A mechanism is truthful **if and only if** the scheduling algorithm is **Monotone**. [Saks & Yu, EC 2005]

Monotonicity

- For any $(t_{i1}, \dots, t_{in})$ and $(t'_{i1}, \dots, t'_{in})$ of machine i, let x and x' be the corresponding assignments, we have

$$\sum_{j=1}^{m} \left(x_{ij} - x'_{ij}\right)\left(t_{ij} - t'_{ij}\right) \leq 0$$

- If job j is assigned to machines i, and we decrease only t_{ij}, then j should still be assigned to i.
  - Otherwise machine i has incentive to lie.

A Truthful Mechanism [Nisan & Ronen, STOC 1999]

- No truthful mechanism has approximation ratio less then 2, even on two machines.
  - Improved to $1 + \sqrt{2} \approx 2.414$ for $m \geq 3$ [Christodoulou et al., SODA 2007]
  - Improved to $1 + \Phi \approx 2.618$ for $m \to \infty$ [Koutsoupias & Vidali, MFCS 2007]

- There exists a truthful mechanism with approximation ratio m.
  - Optimal for two machines.

Lower Bound of 2

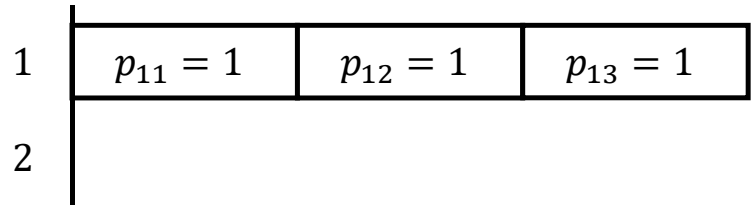
- Monotone ⇔ Truthful ⇒ $t_{ij} = p_{ij}$

- Fix any truthful mechanism on 2 machines, with 3 jobs:

Lower Bound of 2

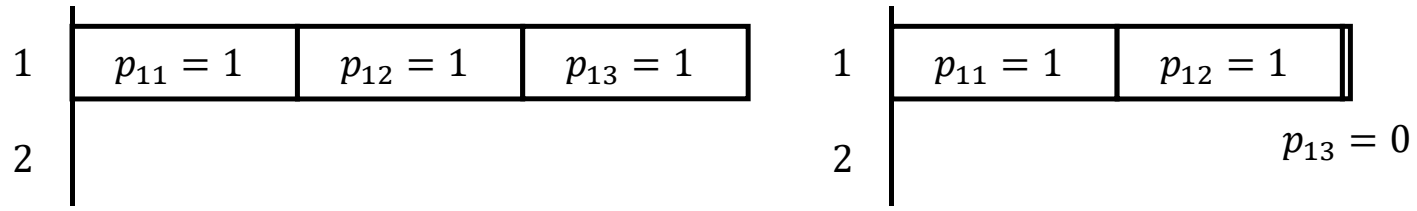
- Monotone ⇔ Truthful ⇒ $t_{ij} = p_{ij}$
- Fix any truthful mechanism on 2 machines, with 3 jobs:
- If $p_{ij} = 1$ for all $i = 1,2$ and $j = 1,2,3$, the scheduler (w.l.o.g.)
  - Schedule all jobs $\{1,2,3\}$ on the first machine, or
  - Schedule $\{1,2\}$ on machine 1 and $\{3\}$ on machine 2.

Case 1: all jobs on machine $\mathbf{1}$



Case 1: all jobs on machine 1

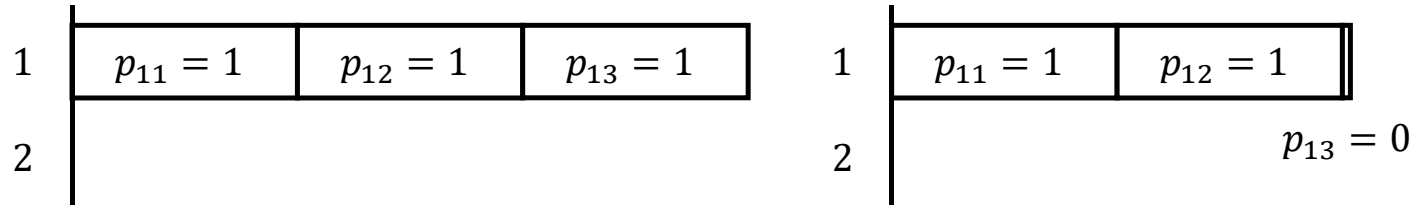
- By monotonicity:
- if machine 1 claims $t_{13} = 0$, without changing other t_{ij}
- then all jobs should still be assigned to machine 1



Case 1: all jobs on machine 1

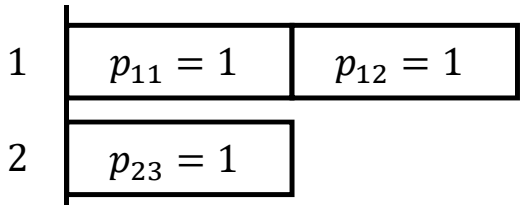
- By monotonicity:
- if machine 1 claims $t_{13} = 0$, without changing other t_{ij}
- then all jobs should still be assigned to machine 1

- For the instance with $p_{13} = 0, p_{ij} = 1$ for other i, j:
- $\mathrm{ALG} = 2, \mathrm{OPT} = 1$



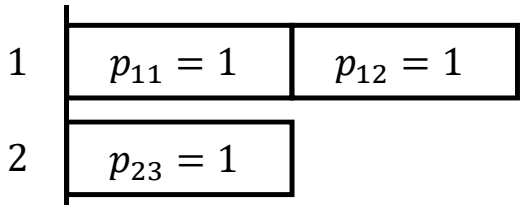
Case 2: jobs $\{1,2\}$ on machine 1

- If machine 2 claims $t_{23} = \epsilon, t_{21} = t_{22} = 1 + \epsilon$,
- by monotonicity: $(x'_{23} - 1) \cdot (1 - \epsilon) \geq \epsilon \cdot (x'_{21} + x'_{22})$



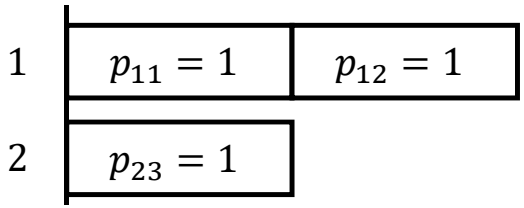
Case 2: jobs $\{1,2\}$ on machine 1

- If machine 2 claims $t_{23} = \epsilon, t_{21} = t_{22} = 1 + \epsilon$,
- by monotonicity: $0 \geq (x'_{23} - 1) \cdot (1 - \epsilon) \geq \epsilon \cdot (x'_{21} + x'_{22})$



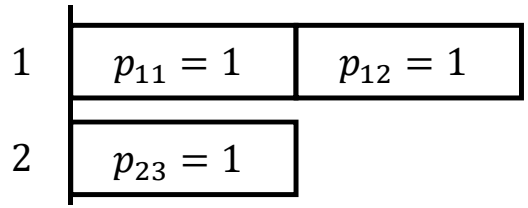
Case 2: jobs $\{1,2\}$ on machine 1

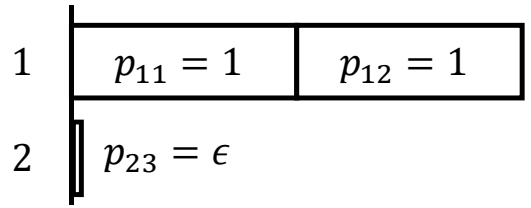
- If machine 2 claims $t_{23} = \epsilon, t_{21} = t_{22} = 1 + \epsilon$,
- by monotonicity: $0 \geq (x'_{23} - 1) \cdot (1 - \epsilon) \geq \epsilon \cdot (x'_{21} + x'_{22}) \geq 0$.



Case 2: jobs $\{1,2\}$ on machine 1

- If machine 2 claims $t_{23} = \epsilon, t_{21} = t_{22} = 1 + \epsilon$,
- by monotonicity: $0 \geq (x'_{23} - 1) \cdot (1 - \epsilon) \geq \epsilon \cdot (x'_{21} + x'_{22}) \geq 0$.
- $\Rightarrow x'_{23} = 1, x'_{21} = x'_{22} = 0$: same schedule.

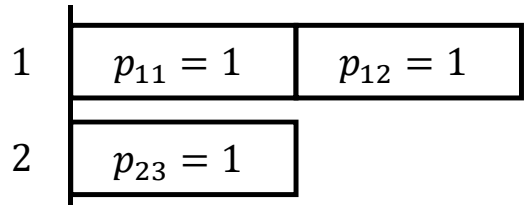


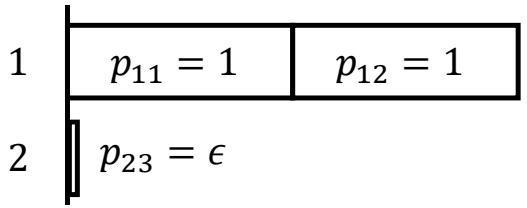


Case 2: jobs $\{1,2\}$ on machine 1

- If machine 2 claims $t_{23} = \epsilon, t_{21} = t_{22} = 1 + \epsilon$,
- by monotonicity: $0 \geq (x'_{23} - 1) \cdot (1 - \epsilon) \geq \epsilon \cdot (x'_{21} + x'_{22}) \geq 0$.
- $\Rightarrow x'_{23} = 1, x'_{21} = x'_{22} = 0$: same schedule.

- For the instance with $p_{23} = \epsilon, p_{21} = p_{22} = 1 + \epsilon$ for $\epsilon \to 0$:
- $\mathrm{ALG} = 2$, $\mathrm{OPT} = 1 + \epsilon$





Upper Bound of m

- Greedy: assign each job to the machine that processes it the fastest: $x_{ij} = 1$ if $t_{ij} < t_{lj}$ for all $l \neq i$.
  - Break tie by machine index

Upper Bound of m

- Greedy: assign each job to the machine that processes it the fastest: $x_{ij} = 1$ if $t_{ij} < t_{lj}$ for all $l \neq i$.
  - Break tie by machine index

- Monotone: $x_{ij}(t_{ij})$ is non-increasing.
  - Assignments of jobs are independent

Upper Bound of m

- Greedy: assign each job to the machine that processes it the fastest: $x_{ij} = 1$ if $t_{ij} < t_{lj}$ for all $l \neq i$.
  - Break tie by machine index

- Monotone: $x_{ij}(t_{ij})$ is non-increasing.
  - Assignments of jobs are independent

- Approximation ratio: $\mathrm{ALG} \leq \sum_j \min_i \{p_{ij}\}$, $\mathrm{OPT} \geq \frac{1}{m} \cdot \sum_j \min_i \{p_{ij}\}$

The payment function

- For every job j assigned to machine i, pay i the smallest claimed processing time from **other** machines.
  - Second Price Auction

- Truthful: for machine i and job j:
  - Suppose the payment is f_{ij}.
  - If $p_{ij} \geq f_{ij}$, then lying (to get the job) is not beneficial.
  - If $p_{ij} < f_{ij}$, then claiming any $t_{ij} < f_{ij}$ gives the same utility.

Randomized Truthful Mechanism

- Truthful in expectation: telling truth maximizes the expected utility
- Universally Truthful: for every realization of the random bits, telling truth maximizes the utility.
  - Since the schedule is random, the approximation ratio can be < 2

- For two machines: (universally truthful)
  - 1.75-approximation [Nisan & Ronen, STOC 1999]
  - Improved to 1.67 [Lu & Yu, STACS 2008]
  - Improved to 1.59 [Lu & Yu, WINE 2008]
- Lower bound of $2 - \frac{1}{m}$ [Mualem & Schapira, SODA 2007]

Selfish Related Machines

Model

- There are n jobs with $a_j = 0$ and $p_j > 0$, no deadline or value.
- Every machine i claims a speed t_i (truth: s_i):
  - processing job j requires $\frac{p_j}{t_i}$ units of time.

Model

- There are n jobs with $a_j = 0$ and $p_j > 0$, no deadline or value.
- Every machine i claims a speed t_i (truth: s_i):
  - processing job j requires $\frac{p_j}{t_i}$ units of time.

- Scheduler assigns jobs to machines, and determines payment f_i for machine i, with the objective of minimizing makespan.

Model

- There are n jobs with $a_j = 0$ and $p_j > 0$, no deadline or value.
- Every machine i claims a speed t_i (truth: s_i):
  - processing job j requires $\frac{p_j}{t_i}$ units of time.

- Scheduler assigns jobs to machines, and determines payment f_i for machine i, with the objective of minimizing makespan.

- Utility of machine i: $f_i(x) - \frac{1}{s_i} \cdot \sum_{j:x_{ij}=1} p_j$

Truthful Mechanisms

- Recall for unrelated machines: approximation: m; hardness: 2 (conjectured to be m) [Nisan & Ronen, STOC 1999]

Truthful Mechanisms

- Recall for unrelated machines: approximation: m; hardness: 2 (conjectured to be m) [Nisan & Ronen, STOC 1999]

- 3-approximate truthful (in expectation) [Archer & Tardos, FOCS 2001]
- PTAS, truthful (in expectation) [Dhangwatnotai et al., FOCS 2008]
- **Deterministic** truthful PTAS [Christodoulou & Kovacs, SODA 2010]
- Deterministic truthful PTAS for maximizing the minimum load, minimizing l_p norm [Epstein et al., SODA 2013]

Selfish Jobs

General Model

- m honest machines.
- Selfish jobs arrive **online**. For each job j:
  - Arrival time a_j
  - Deadline (departure time) d_j
  - Processing time (size) p_{ij} on machine i
  - Value v_j if it is completed
- Assumptions on selfish behaviors:
  - No earlier arrival, later deadline or shorter processing time.

Welfare Maximization Model

- m **identical** machines.
- Selfish jobs arrive **online**. For each job j:
  - Arrival time a_j, deadline d_j, size p_j, value v_j.

- Scheduler: maximize the **total value** of jobs completed.
- Selfish job j: maximize $q_j(x) \cdot v_j - f_j(x)$
  - $q_j(x) \in \{0,1\}$ indicates whether j is completed
  - $f_j(x)$: charging to job j for being processed
  - x: the schedule, which depends on the profiles of jobs

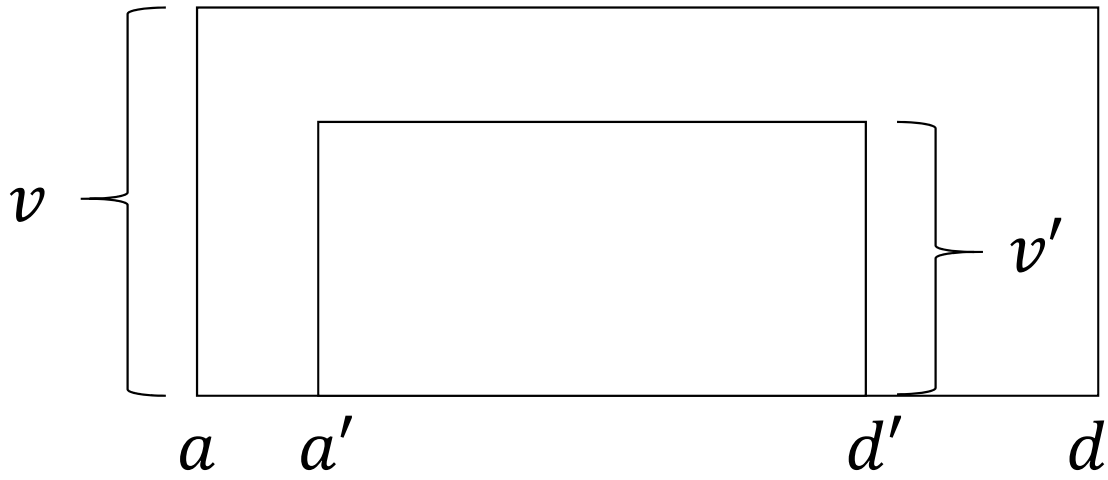
Unit-size Jobs

- Suppose $p_j = 1$ for all job j.
- Suppose a_j, d_j are integers.

- 2-competitive truthful mechanism [Hajiaghayi et al., EC 2005]
  - Greedy algorithm: at the beginning of every time slot, schedule the pending job with maximum value.
  - Greedy is monotone ⇒ truthful.
  - Greedy is 2-competitive.

Dominance and Monotonicity

- (a, d, v) dominates (a', d', v') if $a \leq a' \leq d' \leq d$, and $v > v'$
  - Longer active time window, larger value



Dominance and Monotonicity

- (a, d, v) dominates (a', d', v') if $a \leq a' \leq d' \leq d$, and $v > v'$
  - Longer active time window, larger value

- **Monotone** scheduling algorithm: if job j is scheduled with profile (a', d', v'), then j will be scheduled with profile (a, d, v).

Dominance and Monotonicity

- (a, d, v) dominates (a', d', v') if $a \leq a' \leq d' \leq d$, and $v > v'$
  - Longer active time window, larger value

- **Monotone** scheduling algorithm: if job j is scheduled with profile (a', d', v'), then j will be scheduled with profile (a, d, v).

- Monotone ⇒ truthful (with a carefully defined payment function)

A Simple Payment Function

- For claimed profile (a, d, v), payment function

$$p_j(a, d, v) = q_j(a, d, v) \cdot \min\{v' : q_j(a, d, v') = 1\}$$

- Payment = smallest value that gets the job done.

A Simple Payment Function

- For claimed profile (a, d, v), payment function

$$p_j(a, d, v) = q_j(a, d, v) \cdot \min\{v' : q_j(a, d, v') = 1\}$$

- Payment = smallest value that gets the job done.

- For any a and d, telling truth on v is optimal.
- By monotonicity, lying about a and d is not beneficial.

A Simple Payment Function

- For claimed profile (a, d, v), payment function

$$p_j(a, d, v) = q_j(a, d, v) \cdot \min\{v' : q_j(a, d, v') = 1\}$$

- Payment = smallest value that gets the job done.

- For any a and d, telling truth on v is optimal.
- By monotonicity, lying about a and d is not beneficial.
- Does not apply to randomized/fractional assignments.

General Payment Function

- For claimed profile (a, d, v), payment function

$$f_j(a, d, v) = q_j(a, d, v) \cdot v - \int_0^v q_j(a, d, x)dx$$

General Payment Function

- For claimed profile (a, d, v), payment function

$$f_j(a, d, v) = q_j(a, d, v) \cdot v - \int_0^v q_j(a, d, x)dx$$

- Truthful if and only if (a_j, d_j, v_j) maximizes utility

$$q_j(a, d, v) \cdot v_j - f_j(a, d, v)$$

Suppose otherwise

- Suppose exist $a \geq a_j, d \leq d_j$ and v such that

$$q_j(a, d, v) \cdot v_j - f_j(a, d, v) > q_j\left(a_j, d_j, v_j\right) \cdot v_j - f_j\left(a_j, d_j, v_j\right)$$

Suppose otherwise

- Suppose exist $a \geq a_j, d \leq d_j$ and v such that

$$q_j(a,d,v) \cdot v_j - f_j(a,d,v) > q_j(a_j,d_j,v_j) \cdot v_j - f_j(a_j,d_j,v_j)$$

$$f_j(a_j,d_j,v_j) = q_j(a_j,d_j,v_j) \cdot v_j - \int_0^{v_j} q_j(a_j,d_j,x)dx$$

Suppose otherwise

- Suppose exist $a \geq a_j, d \leq d_j$ and v such that

$$q_j(a, d, v) \cdot v_j - f_j(a, d, v) > \int_0^{v_j} q_j(a_j, d_j, x) dx$$

Suppose otherwise

- Suppose exist $a \geq a_j, d \leq d_j$ and v such that

$$q_j(a,d,v) \cdot v_j - f_j(a,d,v) > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

$$f_j(a,d,v) = q_j(a,d,v) \cdot v - \int_0^{v} q_j(a,d,x)dx$$

Suppose otherwise

- Suppose exist $a \geq a_j, d \leq d_j$ and v such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

Suppose otherwise

- Suppose exist $a \geq a_j, d \leq d_j$ and v such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

Consider three cases: $v = v_j$, $v < v_j$ and $v > v_j$.

If $v = v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v = v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

If $v = v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v = v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

$$\int_0^v q_j(a,d,x)dx > \int_0^v q_j(a_j,d_j,x)dx$$

If $v = v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v = v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

$$\int_0^v q_j(a,d,x)dx > \int_0^v q_j(a_j,d_j,x)dx$$

(a_j, d_j, x) dominates (a, d, x): $q_j(a_j, d_j, x) \geq q_j(a, d, x)$

If $v < v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v < v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

If $v < v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v < v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

$$q_j(a,d,v) \cdot (v_j - v) > \int_v^{v_j} q_j(a_j,d_j,x)dx$$

If $v < v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v < v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

$$q_j(a,d,v) \cdot (v_j - v) > \int_v^{v_j} q_j(a,d,x)dx$$

If $v < v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v < v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

$$q_j(a,d,v) \cdot (v_j - v) > \int_v^{v_j} q_j(a,d,x)dx$$

(a,d,x) dominates (a,d,v): $q_j(a,d,x) \geq q_j(a,d,v)$

If $v > v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v > v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

If $v > v_j$

- Suppose exist $a \geq a_j, d \leq d_j, v > v_j$ such that

$$q_j(a,d,v) \cdot (v_j - v) + \int_0^v q_j(a,d,x)dx > \int_0^{v_j} q_j(a_j,d_j,x)dx$$

$$\int_{v_j}^v q_j(a,d,x)dx > \int_0^v q_j(a,d,x)dx - \int_0^{v_j} q_j(a_j,d_j,x)dx > q_j(a,d,v) \cdot (v - v_j)$$

(a,d,v) dominates (a,d,x): $q_j(a,d,v) \geq q_j(a,d,x)$

General Payment Function

- For claimed profile (a, d, v), payment function

$$f_j(a, d, v) = q_j(a, d, v) \cdot v - \int_0^v q_j(a, d, x)dx$$

- **Truthful** since (a_j, d_j, v_j) maximizes utility

$$q_j(a, d, v) \cdot v_j - f_j(a, d, v)$$

Greedy algorithm is

- Monotone: if not completed, then remains not completed with a later arrival time, earlier deadline and smaller value.

Greedy algorithm is

- Monotone: if not completed, then remains not completed with a later arrival time, earlier deadline and smaller value.

- 2-Competitive: let A be the set of jobs completed by Greedy.
  - Consider any job $j \in \mathrm{OPT} \setminus \mathrm{A}$.
  - Let t_j be the time slot j is processed in OPT.
  - In Greedy, the job processed at t_j has higher value.
  - $v(\mathrm{OPT}) \leq v(\mathrm{A}) + v(\mathrm{OPT} \setminus \mathrm{A}) \leq 2 \cdot v(\mathrm{A})$

Unit-size Jobs (Asynchronous)

- Suppose $p_j = 1$ for all job j.
- Suppose a_j, d_j are non-negative reals.
- Restart/preemption is necessary:
  - Job with tight deadline and large value arrive when all machines are busy

Unit-size Jobs (Asynchronous)

- Suppose $p_j = 1$ for all job j.
- Suppose a_j, d_j are non-negative reals.
- Restart/preemption is necessary:
  - Job with tight deadline and large value arrive when all machines are busy

- 5-competitive truthful mechanism with restart [Hajiaghayi et al., EC 2005]

Unit-size Jobs (Asynchronous)

- Suppose $p_j = 1$ for all job j.
- Suppose a_j, d_j are non-negative reals.
- Restart/preemption is necessary:
  - Job with tight deadline and large value arrive when all machines are busy

- 5-competitive truthful mechanism with restart [Hajiaghayi et al., EC 2005]
- Lower bound of 5 for any restart/preemptive truthful mechanism [Chen et al. IJCAI 2017]

Arbitrary Size Jobs

- Job j has arrival time a_j, deadline d_j, size p_j, and value v_j
- $\omega(1)$ hardness for online preemptive algorithms, when jobs have tight deadlines: $d_j = a_j + p_j$. [Canetti & Irani, SICOMP 1998]
- $O(1)$-competitive preemptive algorithm if the input instance has slack $s = \min_j \left\{ \frac{d_j - a_j}{p_j} \right\} > 1$. [Lucier et al., SPAA 2013]

Arbitrary Size Jobs

- Truthful $O(1)$-competitive algorithm [Azar et al., EC 2015]
  - Slack $s = \min_j \left\{ \frac{d_j - a_j}{p_j} \right\} > 1$.
  - Monotone ⇒ truthful
  - β-Committed (by time $d_j - \beta \cdot p_j$, inform a job j whether it can be completed), given that $s \geq s(\beta)$ is sufficiently large.

Makespan Minimization Model

- Every selfish job j:
  - no deadline or value
  - minimize its own completion time

- Scheduler: minimize the makespan
  - Identical machines
  - Related machines
  - Unrelated machines

Truthful Mechanisms

- Any truthful mechanism without payments?

Truthful Mechanisms

- Any truthful mechanism without payments? **Greedy**!
  - Whenever a job arrives, schedule it to the machine that minimizes its completion time.
  - Break tie (of arrival time) by identity of jobs.
  - Truthful since job utilities are maximized.

Truthful Mechanisms

- Any truthful mechanism without payments? **Greedy**!
  - Whenever a job arrives, schedule it to the machine that minimizes its completion time.
  - Break tie (of arrival time) by identity of jobs.
  - Truthful since job utilities are maximized.

- Competitive ratio of Greedy
  - Identical machines: 2 [Graham, Bell System Technical Journal 1966]
  - Related machines: $\Theta(\log m)$ [Aspnes et al., STOC 1993]
  - Unrelated machines: $\Theta(m)$ [Aspnes et al., STOC 1993]

Better Competitive Ratios?

- Posted price mechanisms:
  - Before the next job arrive, set a price for every machine.
  - Each job decides which machine it goes to, and pays the price.
  - Truthful since jobs make their own decisions.
  - Static price: fixed price for each machine.
  - Dynamic price: prices change depending on current schedule.

Better Competitive Ratios?

- Posted price mechanisms:
  - Before the next job arrive, set a price for every machine
  - Each job decides which machine it goes to, and pays the price
  - Truthful since jobs make their own decisions.
  - Static price: fixed price for each machine
  - Dynamic price: prices change depending on current schedule

- On related machines: $O(1)$-competitive [Feldman et al., EC 2017]
  - By dynamic prices

Constant Competitive Algorithm

- Online algorithm Slow-Fit [Azar et al., Journal of Algorithms 1997]
  - Maintain an upper bound Λ on OPT
  - Schedule every job to the slowest machine without violating the upper bound $2 \cdot \Lambda$.
  - Double Λ otherwise.
  - $O(1)$-competitive.
  - Not truthful

Constant Competitive Algorithm

- Online algorithm Slow-Fit [Azar et al., Journal of Algorithms 1997]
  - Maintain an upper bound Λ on OPT
  - Schedule every job to the slowest machine without violating the upper bound $2 \cdot \Lambda$.
  - Double Λ otherwise.
  - $O(1)$-competitive.
  - Not truthful

- How can we make Slow-Fit truthful?

Constant Competitive Algorithm

- Online algorithm Slow-Fit [Azar et al., Journal of Algorithms 1997]
  - Maintain an upper bound Λ on OPT
  - Schedule every job to the slowest machine without violating the upper bound $2 \cdot \Lambda$.
  - Double Λ otherwise.
  - $O(1)$-competitive.
  - Not truthful

- How can we make Slow-Fit truthful? By dynamic prices.

An illustrating example [Feldman et al., EC 2017]

- Two machines with speeds $s_1 < s_2$
- Suppose currently, machine 1 finishes its jobs at time l_1; machine 2 finishes its jobs at time l_2.
- Current upper bound: Λ

An illustrating example [Feldman et al., EC 2017]

- When a job with processing time p arrives
  - Slow-Fit assigns the job to machines 1 if $l_1 + \frac{p}{s_1} \leq 2 \cdot \Lambda$
  - Set price π (**independent of p**) on machine 2 such that

$$l_1 + \frac{p}{s_1} \leq l_2 + \frac{p}{s_2} + \pi \Leftrightarrow l_1 + \frac{p}{s_1} \leq 2 \cdot \Lambda$$

- Set $\pi = l_1 - l_2 + \left(1 - \frac{s_1}{s_2}\right) \cdot (2 \cdot \Lambda - l_1)$

General Algorithm

- By dynamic prices on machines:
- Selfish behavior of jobs results in Slow-Fit?

General Algorithm

- By dynamic prices on machines:
- Selfish behavior of jobs results in ~~Slow-Fit?~~

General Algorithm

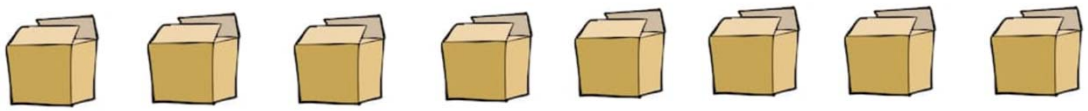
- By dynamic prices on machines:
- Selfish behavior of jobs results in ~~Slow-Fit?~~

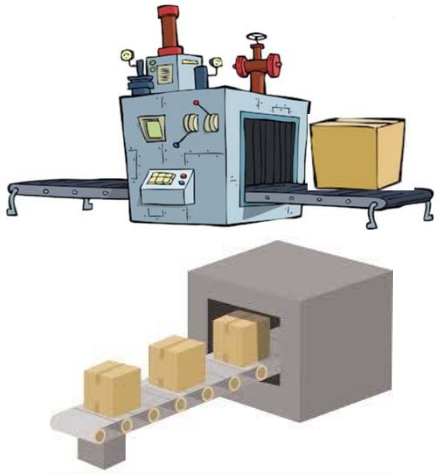
- Selfish behavior of jobs results in **Flex-Fit**, an algorithm similar to Slow-Fit, but has a higher flexibility on job assignments.
  - The competitive ratio of Flex-Fit is only a constant times larger than that of Slow-Fit. [Feldman et al., EC 2017]

Selfish Machines and Selfish Jobs

Classic Online Load Balancing Problem

- M: a set of m related machines, each machine i has speed s_i;
- N: a set of n online jobs, each job j has size p_j;
- Solution: Process all jobs on the machines without preemption and migration;
- Goal: Minimize the maximum finishing time over all machines (makespan).





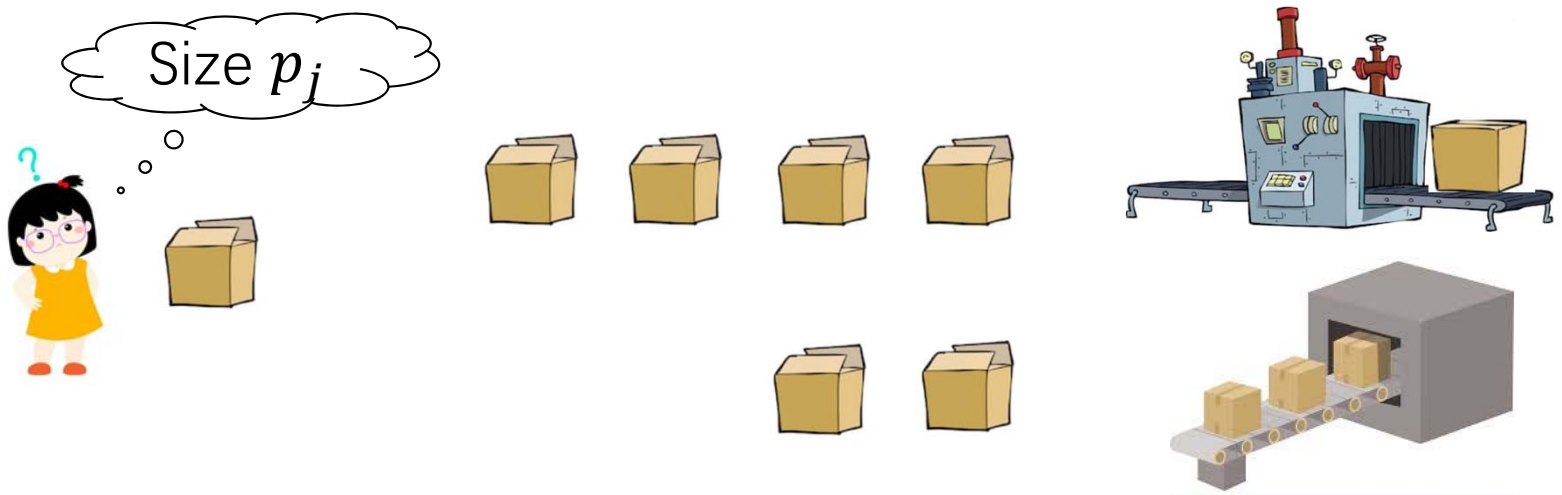
Classic Online Load Balancing Problem

- M: a set of m related machines, each machine i has speed $\mathbf{s}_i$;
- N: a set of n online jobs, each job j has size p_j;
- Solution: Process all jobs on the machines without preemption and migration;
- Goal: Minimize the maximum finishing time over all machines (makespan).

- Has a long history [Graham69, FW00, BCK00, ⋯]

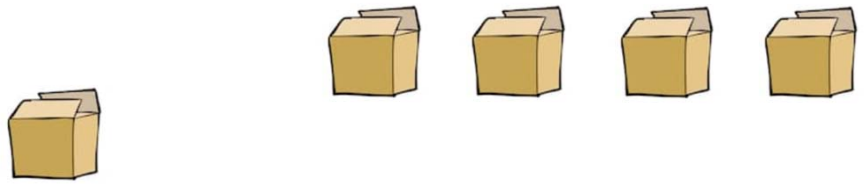
Facts in Multi-agent System

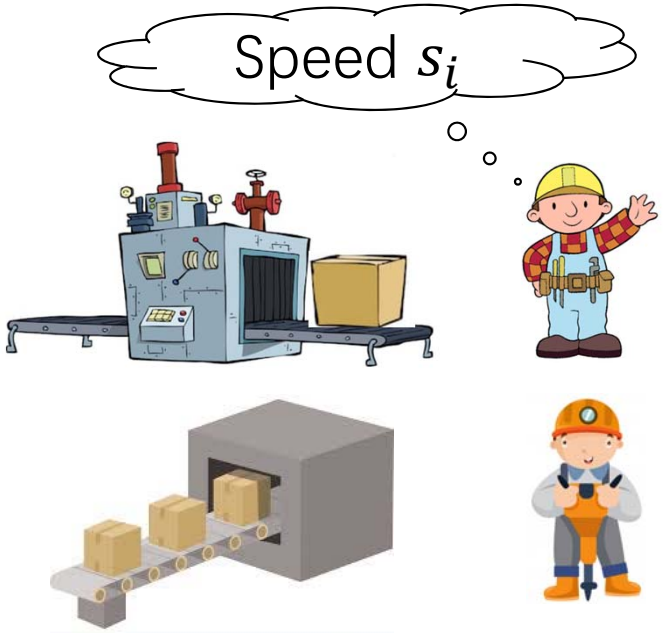
1. The jobs are controlled by strategic agents:
   - The size is only known to the agent and her goal is to minimize her own cost (finishing time + potential payment).
   - Truthful mechanisms against strategic jobs [AAF'97, FFR'17, ⋯]



Facts in Multi-agent System

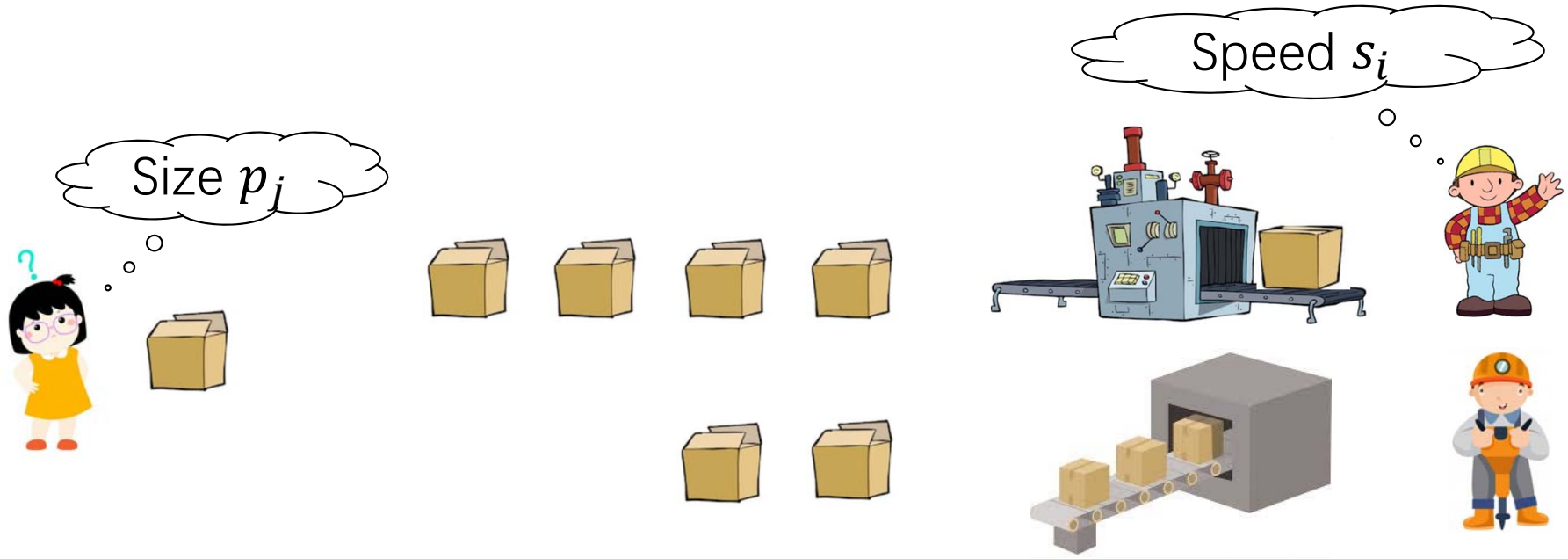
2. The machines are controlled by strategic agents:
   - From the view of incentive compatibility:
     The assignment and payment should be truthful.
     [AAS'05, Kovacs'05, CK'13, ELS'16]
   - From the view of stability:
     The assignment should be fair.
     [DPR'11, SR'13]





Our Problem

- Both jobs and machines are controlled by strategic agents.



Target Mechanisms

- Truthful against strategic jobs:
  - It is each job's best strategy to report her true size.
- Well-behaved for machines ($\approx$ weighted Proportionality):
  - A machine has workload no smaller than that of any slower machine.
- Why well-behaved mechanisms?
  - Guarantees the fairness among the machines (stable cooperation);
  - Implies truthfulness of machines in some cases.
- Benchmark: Optimal offline makespan.

To what extent the makespan can be minimized under a well behaved mechanism that is truthful for strategic jobs?

Hardness of well-behavior

Hardness of well-behaved schedules

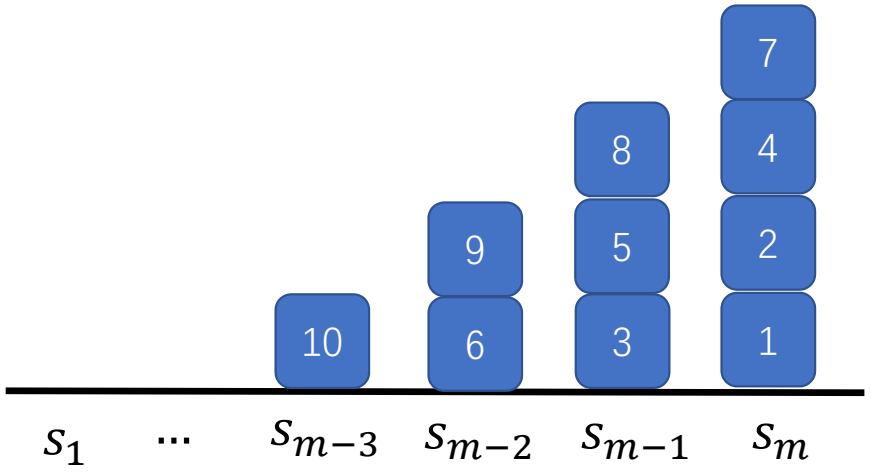
Theorem 1. *Well behaved schedules have competitive ratios* $\Omega(\sqrt{m})$.

- "Proof ":
  - A set of m machines with speeds:

$$s_1 = 1 + \frac{1}{2^{m-1}},$$
$$s_2 = 1 + \frac{1}{2^{m-2}},$$
$$\vdots$$
$$s_m = 2.$$

  - A set of m online jobs with sizes $p_j = s_j$.
  - $OPT = 1$ by assigning job i to machine i.

- Bad case: the jobs arrive in increasing order of sizes.



Our Mechanism

Almost Well-behaved Algorithms

- Strong lower bound → relax well-behavior
- γ-Almost well-behavior: a machine i has no smaller workload than machine i' only if $s_i \geq \gamma s_{i'}$.
- In our paper, we set $\gamma = 2$.

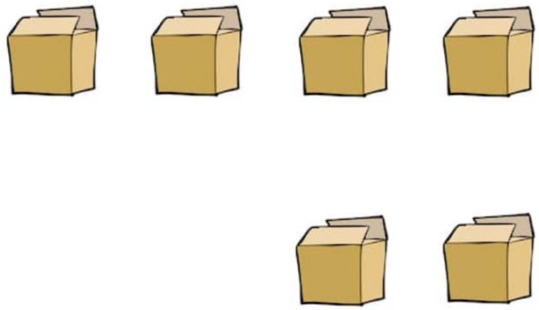
- Preprocessing:
  - Rounding each speed s_i down to the largest power of 2.
  - $s_1 \leq s_2 \leq \cdots \leq s_n$ and each s_i is some power of 2.
- Next:
  - Design a well-behaved mechanism on rounded speeds that is truthful to jobs.

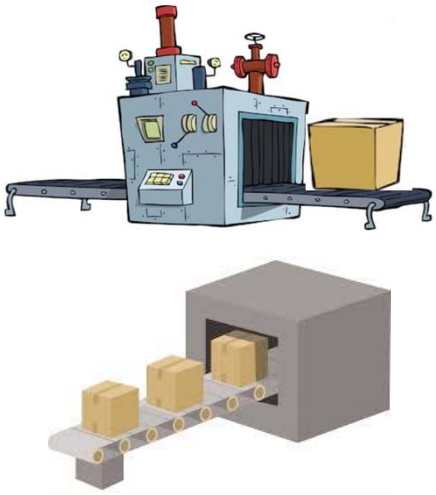
Dynamic Posted Pricing Mechanism

- Posted Prices of Machines:

Machine i: $price_i$ and current $workload_i$

The job can select the machine which minimizes her cost (finishing time + payment).





Dynamic Posted Pricing Mechanism

- Posted Prices of Machines:
  - Inherently truthful: if the prices are determined irrespective of the size of the next coming job;
  - Simple:
    - Jobs need not trust nor understand the logic underlying the mechanism
    - Jobs are not required to reveal their sizes (agents may not be aware of her true type).

- Difficulty:
  - Designing prices such that the selections of jobs imply well-behavior.

Dynamic Posted Pricing Mechanism, cont.

Multiple machines with the same speed?

Set the prices to be infinity except the one with minimum makespan.

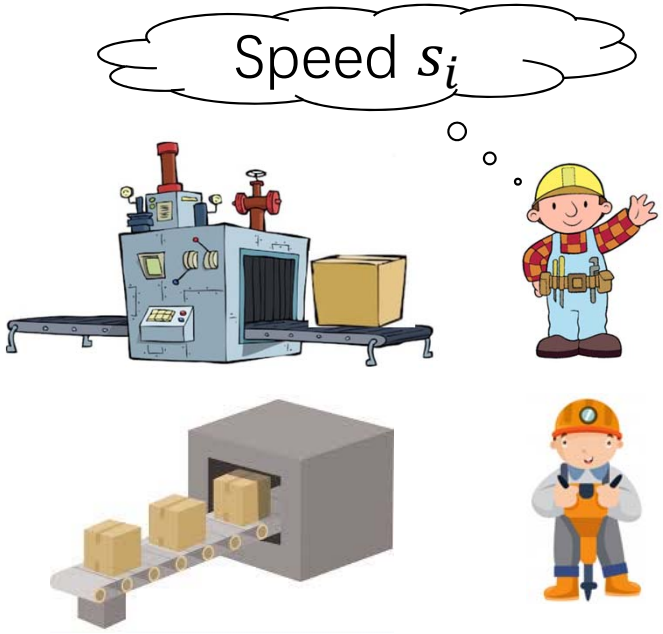
- Assume all speeds s_i are different: $s_1 < s_2 < \cdots < s_n$.
- In each round, a job j arrives:
  - Let $C_i(j)$ be the makespan of machine i before j arrives.
  - Let $\pi_m = 0$, and for each $i < m$, $\pi_i := \frac{s_i}{s_{i+1}}(C_{i+1}(j) - C_i(j))$.
  - Let $\rho_i(j) = \sum_{l \geq i} \pi_i$ be the price of machine i before j arrives.
  - So j will select the machine i such that $C_i(j) + \frac{p_i}{s_i} + \rho_i(j)$ is minimized.
  - Resulting schedule is well-behaved.

Theorem 2. *The mechanism is $O(\log m)$-competitive in general and $O(1)$-competitive if the sizes of online jobs are bounded.*

Corollaries

Corollary. *Combining Myerson's Lemma, our mechanism is also truthful for the machines if*

1. *all jobs have unit size; or*
2. $m = 2$, *i.e., two machines.*



Conclusion

Conclusion

- We show that any well-behaved mechanism must have a competitive ratio $\Omega(\sqrt{m})$.
- We propose an almost well-behaved dynamic posted-price mechanism that is truthful for the jobs and has a competitive ratio of $O(\log m)$.
- We show that when all jobs have unit size or there are only two machines, our mechanism is actually truthful against strategic machines.

Future Direction

Truthful mechanism for both the strategic jobs and strategic machines.

- A naïve mechanism (VCG):
  - Let $s_1 \leq \cdots \leq s_{m-1} \leq s_m$ be the reported speeds;
  - Machine m gets all jobs with payment $\frac{1}{p_{m-1}} \sum_{j \in J} s_j$;
  - $ALG \leq m \cdot OPT$.

- We would like to see better mechanisms.