On Some Stochastic Load Balancing Problems

Anupam Gupta

Carnegie Mellon University

Joint work with
Amit Kumar, IIT Delhi
Viswanath Nagarajan, Michigan
Xiangkun Shen, Michigan

(appeared at SODA 2018)

Optimization under Uncertainty

- Question: How to model/solve problems with uncertainty in input/actions?
  - data not yet available, or obtaining exact data difficult/expensive
  - actions have uncertainty in outcomes
  - (my talk) we only have *stochastic* predictions

Optimization under Uncertainty

- Question: How to model/solve problems with uncertainty in input/actions?
  - data not yet available, or obtaining exact data difficult/expensive
  - actions have uncertainty in outcomes
  - (my talk) we only have *stochastic* predictions

Goal: get algos making (near)-optimal decisions given predictions.

Optimization under Uncertainty

- Question: How to model/solve problems with uncertainty in input/actions?
  - data not yet available, or obtaining exact data difficult/expensive
  - actions have uncertainty in outcomes
  - (my talk) we only have *stochastic* predictions

Goal: get algos making (near)-optimal decisions given predictions.

- worst-case analysis (vs. queueing perspective, cf. Mor's talk)
  - Relate to performance of best strategy *on worst-case instance*

Optimization under Uncertainty

- Question: How to model/solve problems with uncertainty in input/actions?
  - data not yet available, or obtaining exact data difficult/expensive
  - actions have uncertainty in outcomes
  - (my talk) we only have *stochastic* predictions

Goal: get algos making (near)-optimal decisions given predictions.

- worst-case analysis (vs. queueing perspective, cf. Mor's talk)
  - Relate to performance of best strategy *on worst-case instance*
- given predictions (vs. all-adversarial model as in competitive analysis)

Approximation Algorithms for Stochastic Optimization

- High Level Model:
  - Inputs random variables $X_1, X_2, \ldots$ with known distributions
  - (for today) assume discrete distributions, explicit access to them

Approximation Algorithms for Stochastic Optimization

- High Level Model:
  - Inputs random variables $X_1, X_2, \ldots$ with known distributions
  - (for today) assume discrete distributions, explicit access to them
  - outcomes not known a priori, revealed over time
  - want to optimize, say, $\mathbb{E}[\textit{objective}]$.

Approximation Algorithms for Stochastic Optimization

- High Level Model:
  - Inputs random variables $X_1, X_2, \ldots$ with known distributions
  - (for today) assume discrete distributions, explicit access to them
  - outcomes not known a priori, revealed over time
  - want to optimize, say, $\mathbb{E}[\textit{objective}]$.
  - many different models
    - e.g., adaptive vs. non-adaptive
    - e.g., single-stage vs. multi-stage

Approximation Algorithms for Stochastic Optimization

- High Level Model:
  - Inputs random variables $X_1, X_2, \ldots$ with known distributions
  - (for today) assume discrete distributions, explicit access to them
  - outcomes not known a priori, revealed over time
  - want to optimize, say, $\mathbb{E}[\textit{objective}]$.
  - many different models
    - e.g., adaptive vs. non-adaptive
    - e.g., single-stage vs. multi-stage
- Today: centralized load-balancing problem, minimizing $\mathbb{E}[\textit{makespan}]$.

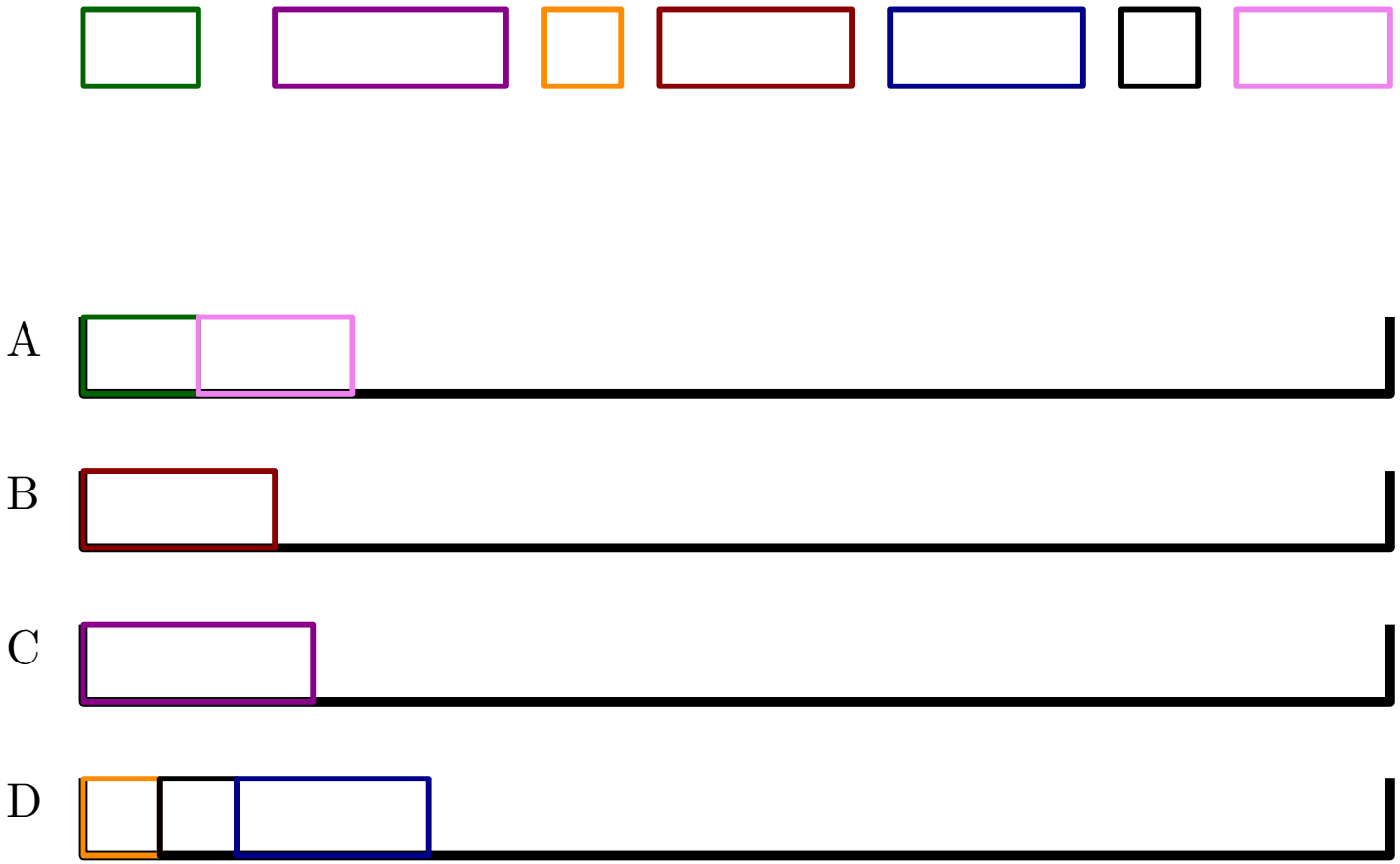
today's problem

(Classical) Load Balancing Problem

- Schedule n jobs on m machines to minimize *makespan*.
- Graham list-scheduling from 1966.

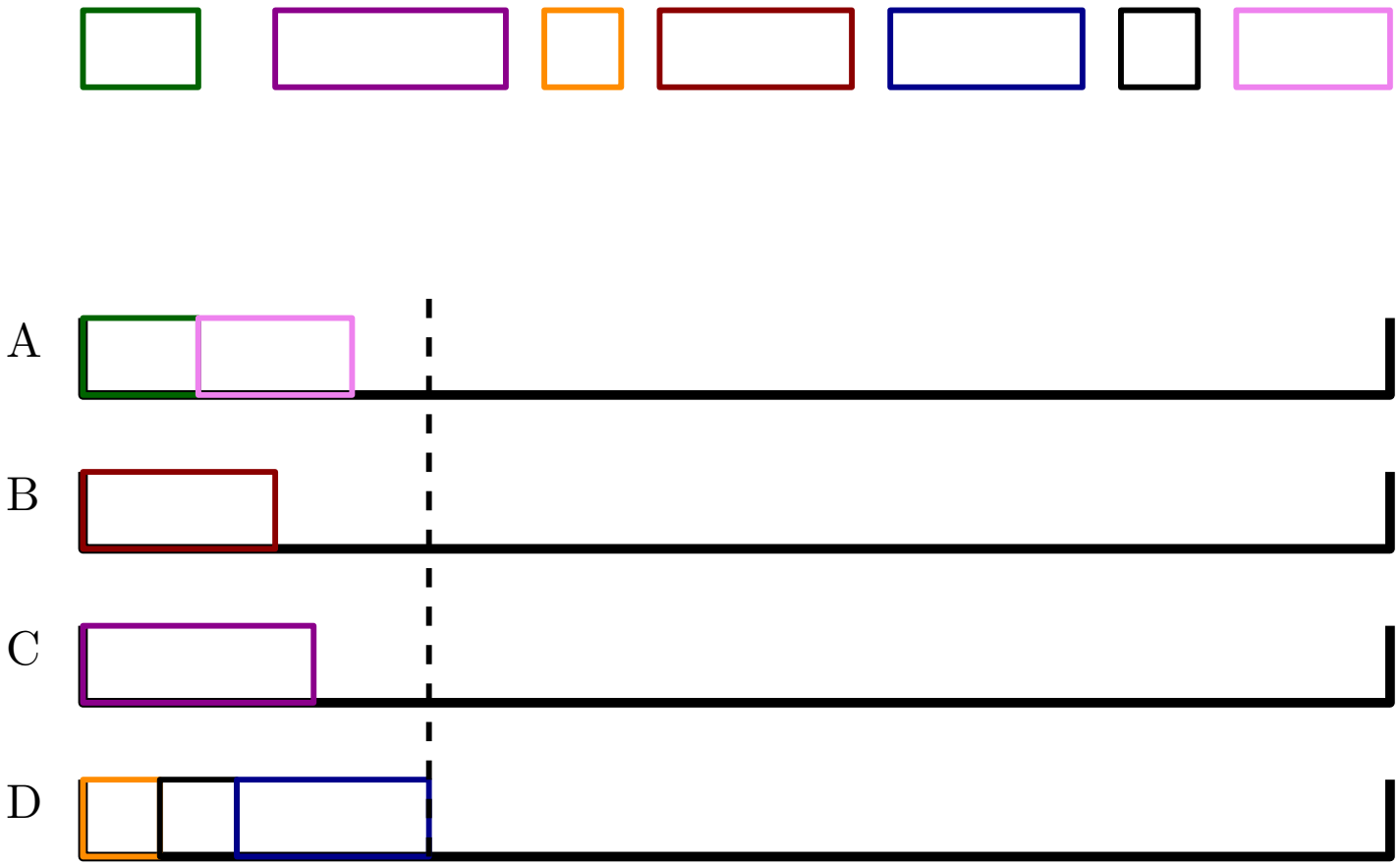
(Classical) Load Balancing Problem

- Schedule n jobs on m machines to minimize *makespan*.
- Graham list-scheduling from 1966.



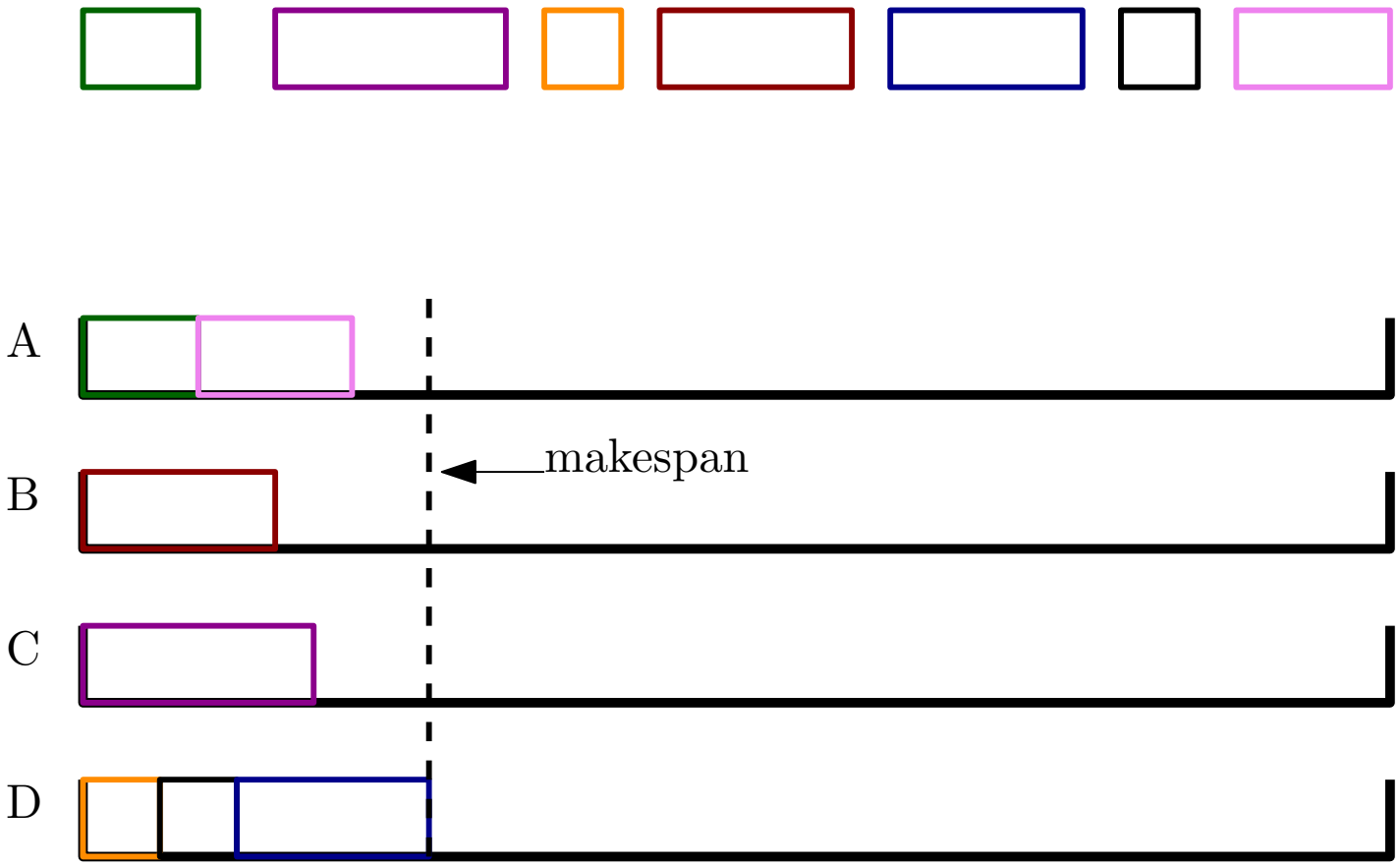
(Classical) Load Balancing Problem

- Schedule n jobs on m machines to minimize *makespan*.
- Graham list-scheduling from 1966.



(Classical) Load Balancing Problem

- Schedule n jobs on m machines to minimize *makespan*.
- Graham list-scheduling from 1966.



much work on the deterministic problem

- Simplest Model: Identical machines:
  - List Scheduling: 2-approx [Graham 66], PTAS [Hochbaum, Shmoys 87].

much work on the deterministic problem

- Simplest Model: Identical machines:
  - List Scheduling: 2-approx [Graham 66], PTAS [Hochbaum, Shmoys 87].
- Most General: Unrelated machines:
  - Jobs have different sizes on different machines.

much work on the deterministic problem

- Simplest Model: Identical machines:
  - List Scheduling: 2-approx [Graham 66], PTAS [Hochbaum, Shmoys 87].
- Most General: Unrelated machines:
  - Jobs have different sizes on different machines.
  - 2-approx [Lenstra, Shmoys, Tardos 90], better for special cases.

Stochastic Load Balancing

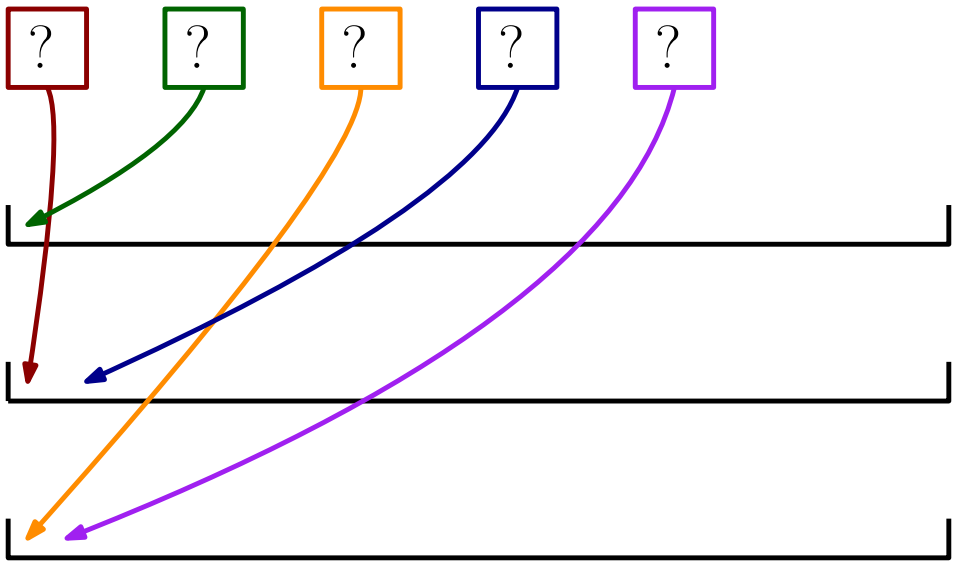
- Job j on machine i takes on size X_{ij} (r.v. with known distribution)
- Today: these r.v.s are independent
- Find an assignment to minimize *expected* makespan:

$$\mathbb{E}\left[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}\right].$$

Stochastic Load Balancing

- Job j on machine i takes on size X_{ij} (r.v. with known distribution)
- Today: these r.v.s are independent
- Find an assignment to minimize *expected* makespan:

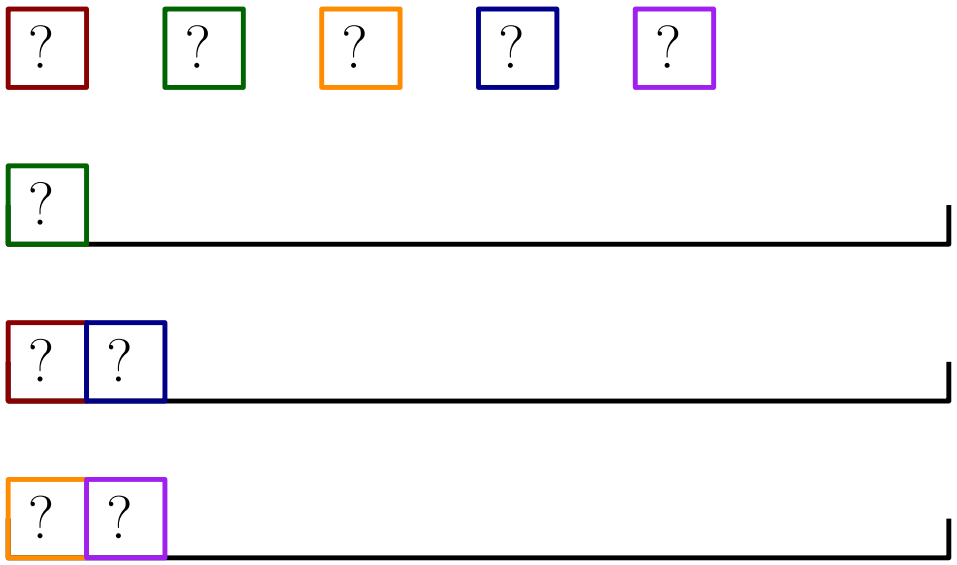
$$\mathbb{E}\left[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}\right].$$



Stochastic Load Balancing

- Job j on machine i takes on size X_{ij} (r.v. with known distribution)
- Today: these r.v.s are independent
- Find an assignment to minimize *expected* makespan:

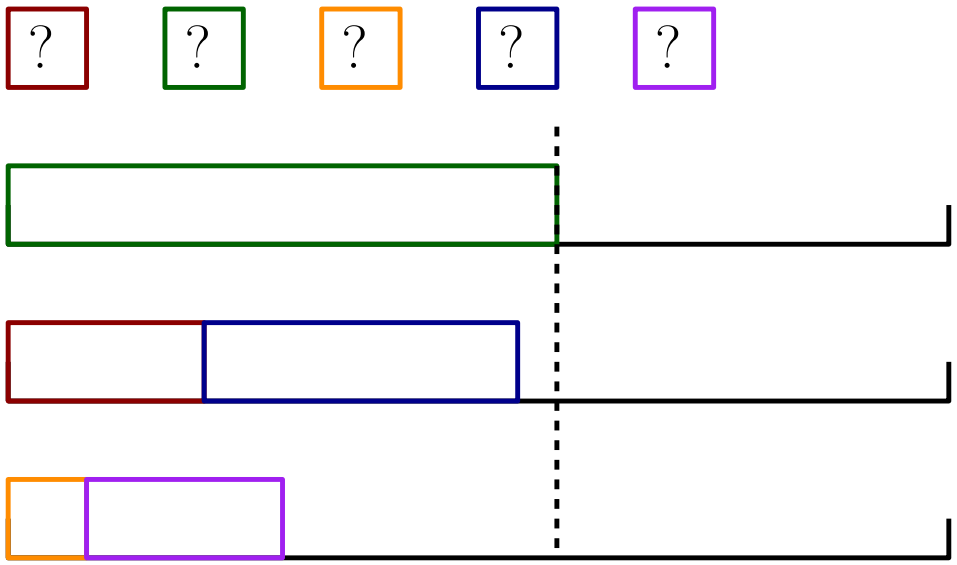
$$\mathbb{E}\left[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}\right].$$



Stochastic Load Balancing

- Job j on machine i takes on size X_{ij} (r.v. with known distribution)
- Today: these r.v.s are independent
- Find an assignment to minimize *expected* makespan:

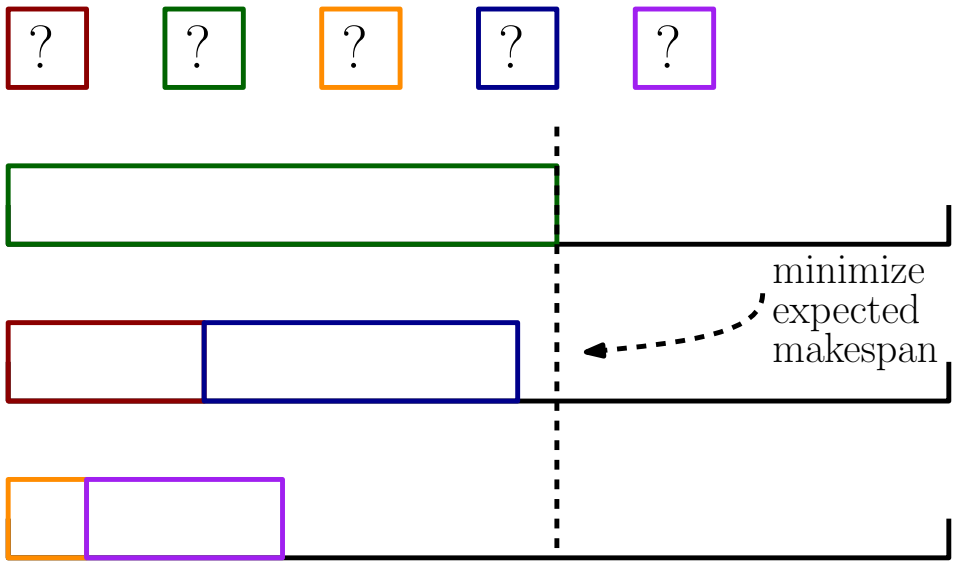
$$\mathbb{E}\left[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}\right].$$



Stochastic Load Balancing

- Job j on machine i takes on size X_{ij} (r.v. with known distribution)
- Today: these r.v.s are independent
- Find an assignment to minimize *expected* makespan:

$$\mathbb{E}\left[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}\right].$$



Related Work: Stochastic Job Sizes

- #P-hard to evaluate objective exactly
- $O(1)$-approximation for *identical machines* [Kleinberg, Rabani, Tardos 00].
- Better results for special classes of job size distributions [Goel, Indyk 99].
  - Poisson distributed job sizes: 2-approximation.
  - Exponential distributed job sizes: PTAS.

Related Work: Stochastic Job Sizes

- #P-hard to evaluate objective exactly
- $O(1)$-approximation for *identical machines* [Kleinberg, Rabani, Tardos 00].
- Better results for special classes of job size distributions [Goel, Indyk 99].
  - Poisson distributed job sizes: 2-approximation.
  - Exponential distributed job sizes: PTAS.
- What about general unrelated case?

Main Result

Theorem

An $O(1)$-approx algo for minimizing $\mathbb{E}[\textit{makespan}]$ on **unrelated** *machines.*

Roadmap and Assumptions

- Identical machines case
- Ideas needed for unrelated machines (and sketch of proof)

Roadmap and Assumptions

- Identical machines case
- Ideas needed for unrelated machines (and sketch of proof)

- By scaling, assume $\mathbb{E}[OPTmakespan] = 1$.

Roadmap and Assumptions

- Identical machines case
- Ideas needed for unrelated machines (and sketch of proof)

- By scaling, assume $\mathbb{E}[OPTmakespan] = 1$.
- Assume each job is "small": $\Pr[size > E[OPTmakespan]] = 0$.
  - Easy extension to general sizes

Deterministic Surrogate

- Find deterministic quantity as a surrogate for each r.v.
- Do optimization over these deterministic quantities
- Surrogate = expected size?

Deterministic Surrogate

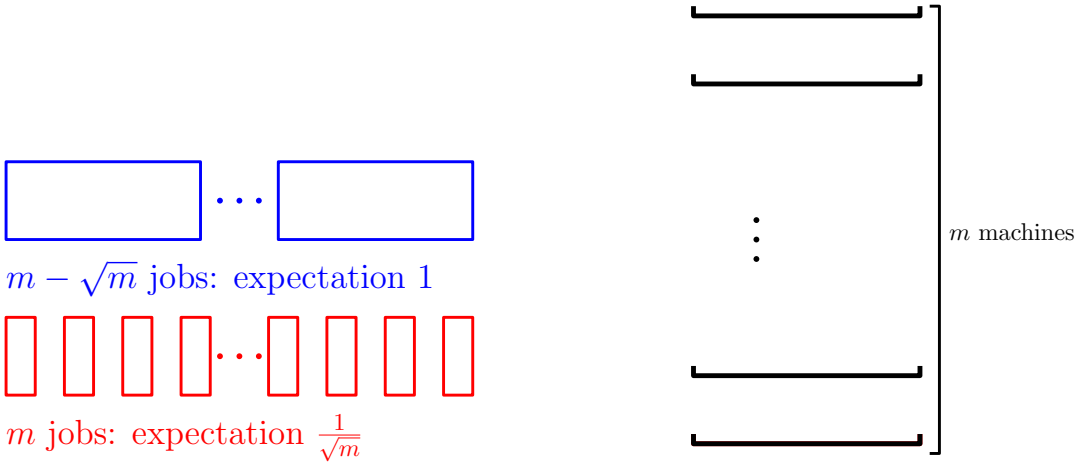
- Find deterministic quantity as a surrogate for each r.v.
- Do optimization over these deterministic quantities
- Surrogate = expected size? **(No!)**

Deterministic Surrogate

- Find deterministic quantity as a surrogate for each r.v.
- Do optimization over these deterministic quantities
- Surrogate = expected size? **(No!)**

Bad Example

- Type 1: size 1 (deterministic).
- Type 2: size Bernoulli$(0, 1)$ r.v. with $p = \frac{1}{\sqrt{m}}$.

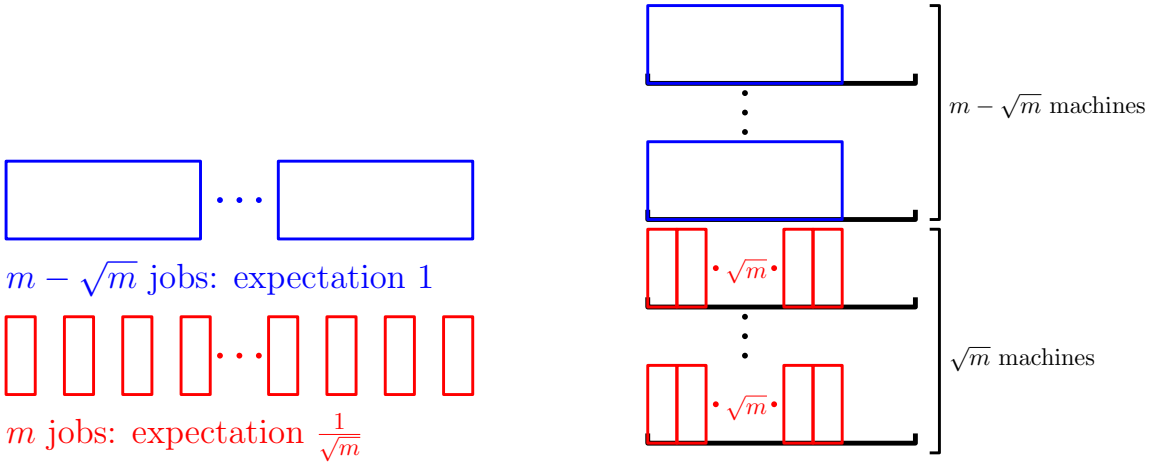


Deterministic Surrogate

- Find deterministic quantity as a surrogate for each r.v.
- Do optimization over these deterministic quantities
- Surrogate = expected size? **(No!)**

Bad Example

- Type 1: size 1 (deterministic).
- Type 2: size Bernoulli$(0, 1)$ r.v. with $p = \frac{1}{\sqrt{m}}$.

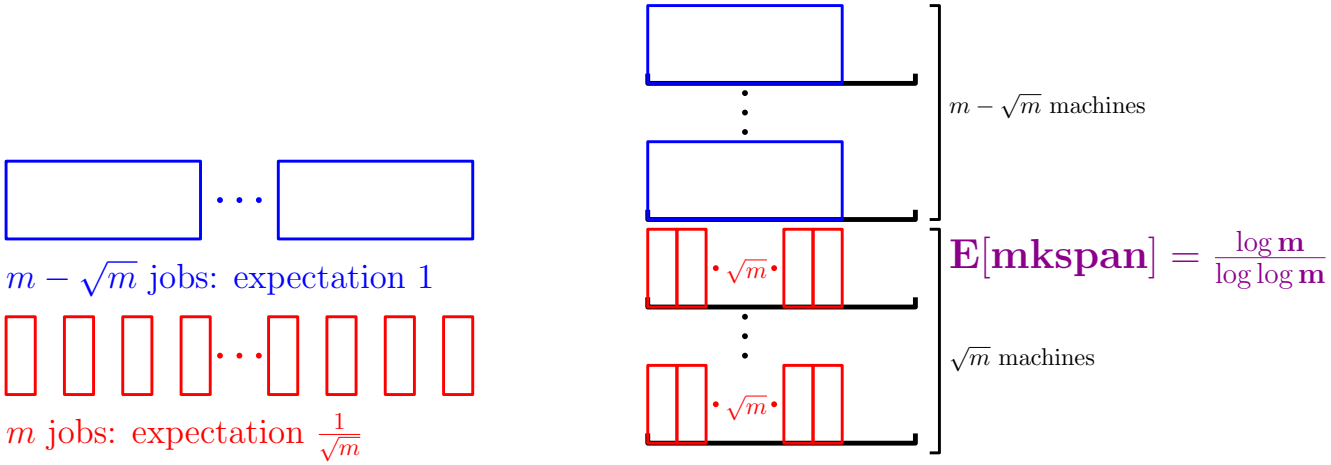


Deterministic Surrogate

- Find deterministic quantity as a surrogate for each r.v.
- Do optimization over these deterministic quantities
- Surrogate = expected size? **(No!)**

Bad Example

- Type 1: size 1 (deterministic).
- Type 2: size Bernoulli$(0, 1)$ r.v. with $p = \frac{1}{\sqrt{m}}$.

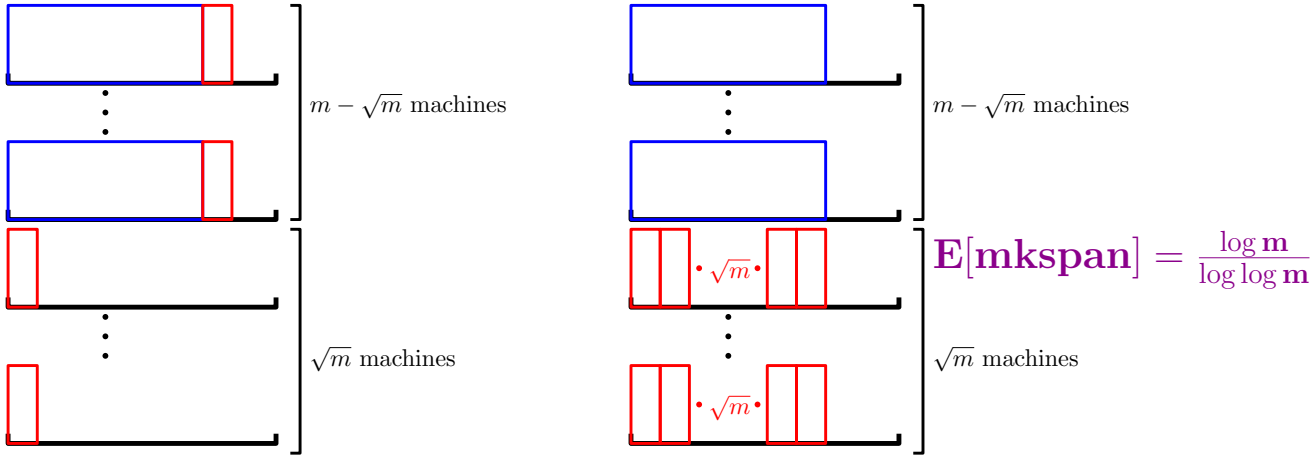


Deterministic Surrogate

- Find deterministic quantity as a surrogate for each r.v.
- Do optimization over these deterministic quantities
- Surrogate = expected size? **(No!)**

Bad Example

- Type 1: size 1 (deterministic).
- Type 2: size Bernoulli$(0, 1)$ r.v. with $p = \frac{1}{\sqrt{m}}$.

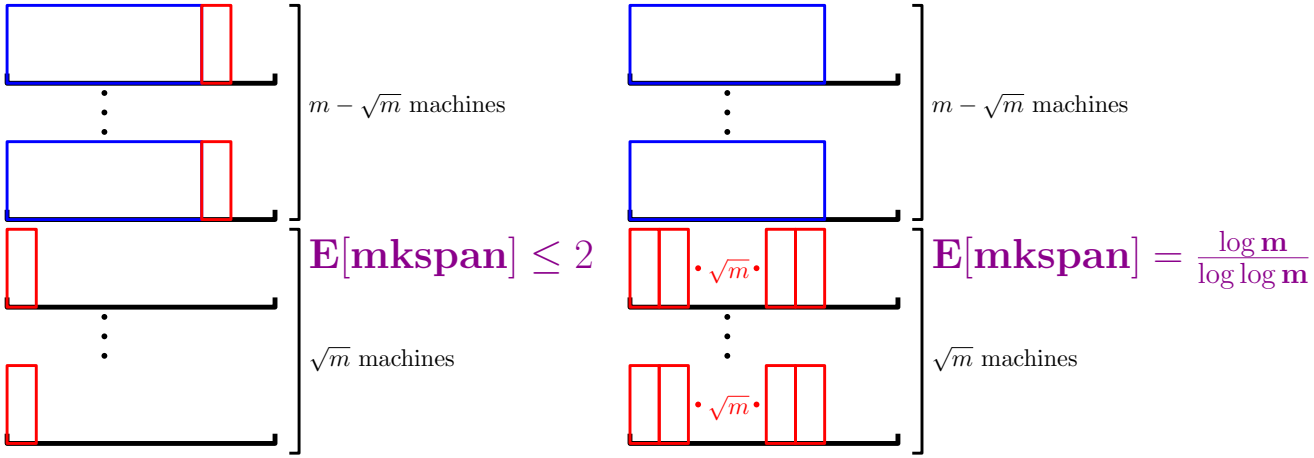


Deterministic Surrogate

- Find deterministic quantity as a surrogate for each r.v.
- Do optimization over these deterministic quantities
- Surrogate = expected size? **(No!)**

Bad Example

- Type 1: size 1 (deterministic).
- Type 2: size Bernoulli$(0, 1)$ r.v. with $p = \frac{1}{\sqrt{m}}$.



Our Chief Weapon: **Effective Size**

Effective Size

[Hui 88] [Elwalid, Mitra 93] [Kelly 96]

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$

Effective Size

[Hui 88] [Elwalid, Mitra 93] [Kelly 96]

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right] = \frac{1}{\log k} \cdot \log \mathbb{E}\left[k^X\right].$$

Effective Size

[Hui 88] [Elwalid, Mitra 93] [Kelly 96]

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right] = \frac{1}{\log k} \cdot \log \mathbb{E}\left[k^X\right].$$

- E.g., $X = C$ w.p. 1, then $\beta_k(X) = C$.

Effective Size

[Hui 88] [Elwalid, Mitra 93] [Kelly 96]

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right] = \frac{1}{\log k} \cdot \log \mathbb{E}\left[k^X\right].$$

- E.g., $X = C$ w.p. 1, then $\beta_k(X) = C$.
- E.g., $X = \mathit{Bernoulli}(1/\sqrt{k})$, then $\beta_k(X) = \frac{\log(1/\sqrt{k} \cdot k^1 + (1 - 1/\sqrt{k}) \cdot k^0)}{\log k} \geq 1/2$.

Effective Size

[Hui 88] [Elwalid, Mitra 93] [Kelly 96]

For any random variable X and parameter $k > 1$, define

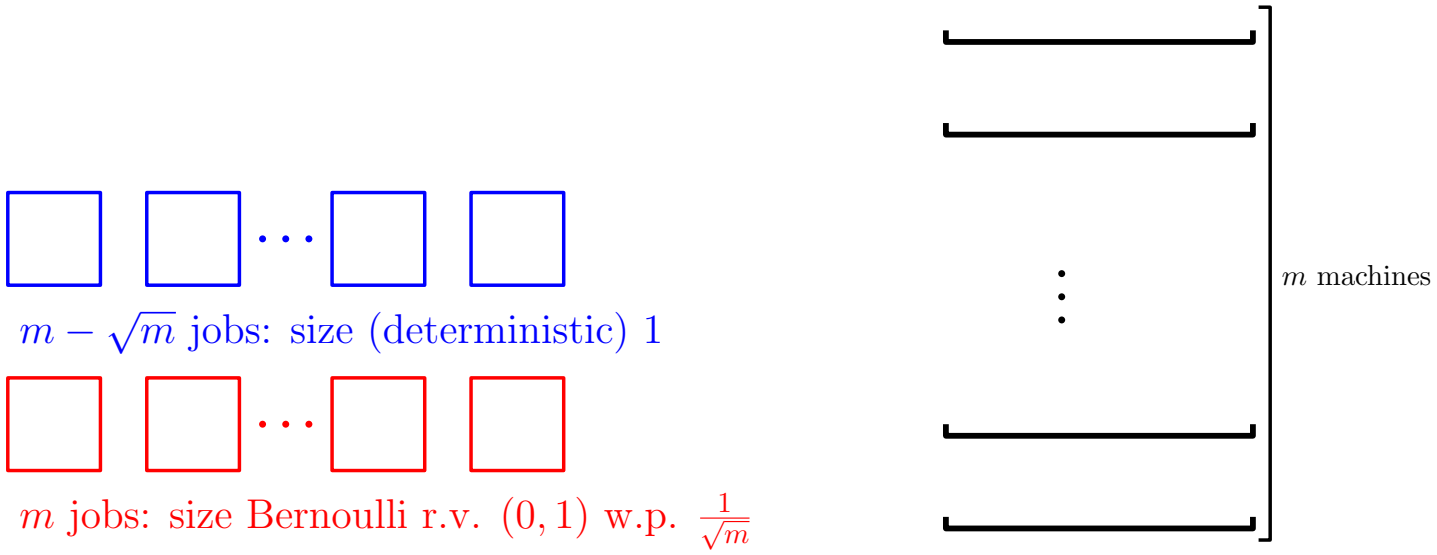
$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right] = \frac{1}{\log k} \cdot \log \mathbb{E}\left[k^X\right].$$

- E.g., $X = C$ w.p. 1, then $\beta_k(X) = C$.
- E.g., $X = \mathit{Bernoulli}(1/\sqrt{k})$, then $\beta_k(X) = \frac{\log(1/\sqrt{k} \cdot k^1 + (1 - 1/\sqrt{k}) \cdot k^0)}{\log k} \geq 1/2$.
- Increasing function of k

Effective Size

For any random variable X and parameter $k > 1$, define

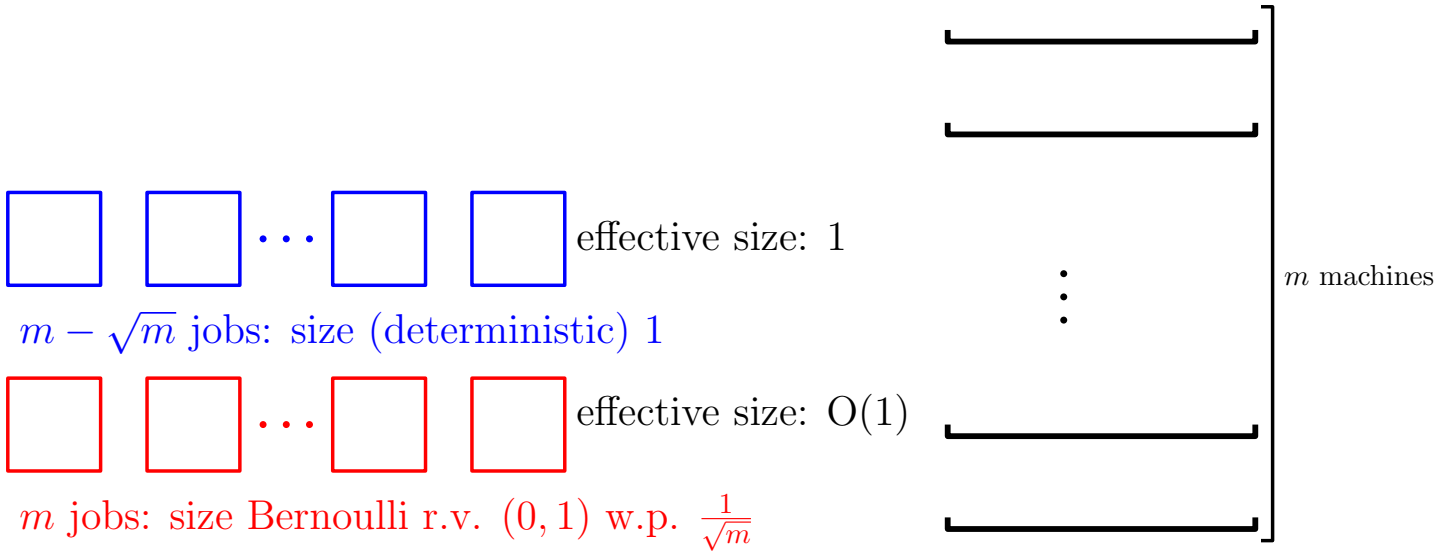
$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$



Effective Size

For any random variable X and parameter $k > 1$, define

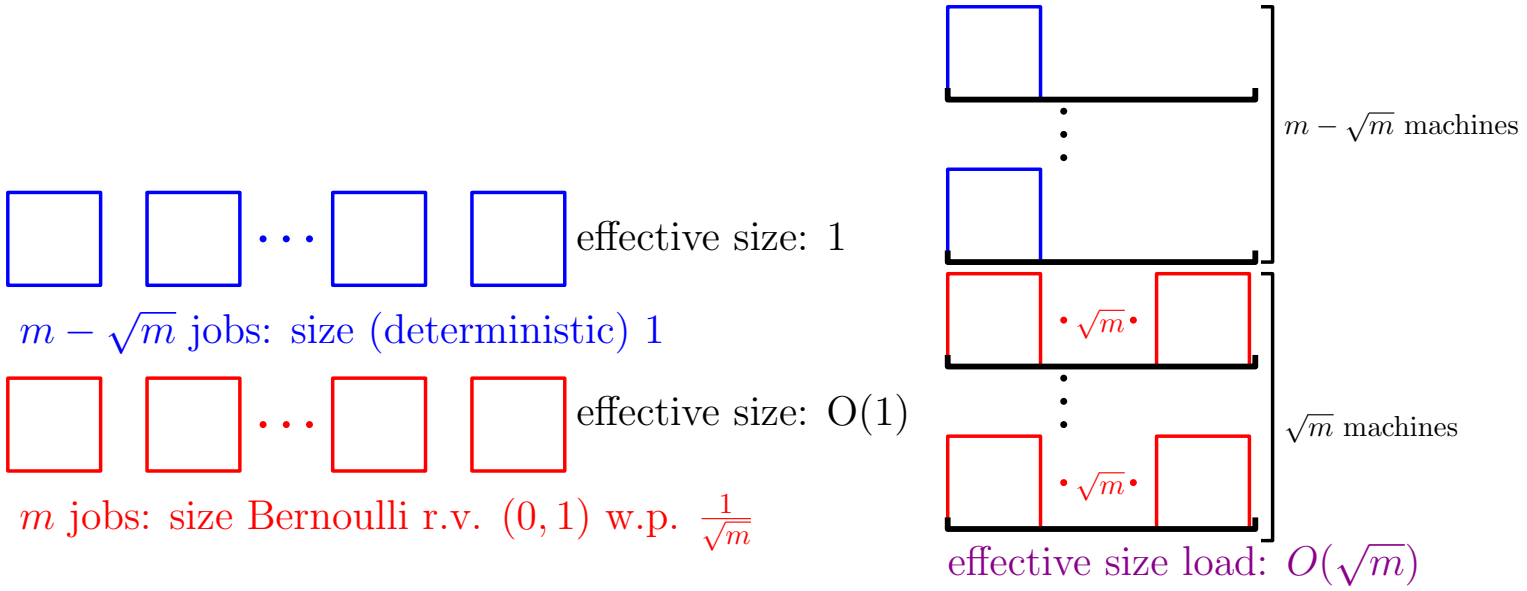
$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$



Effective Size

For any random variable X and parameter $k > 1$, define

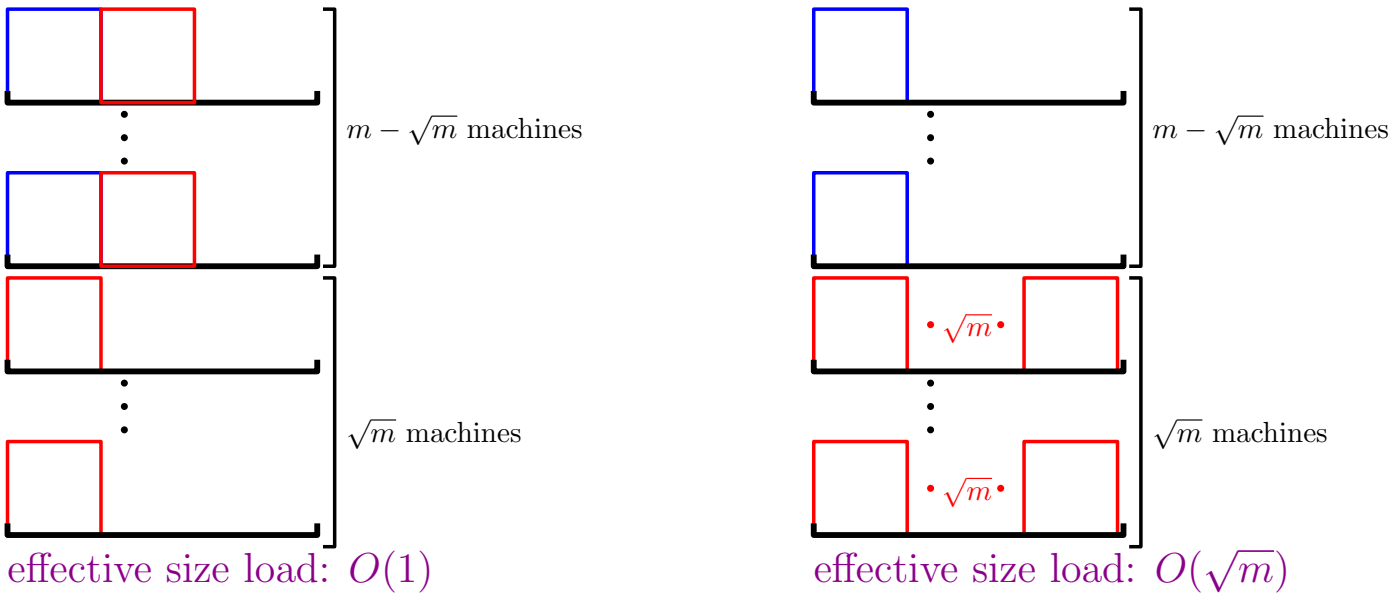
$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$



Effective Size

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$



Effective Size

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$

Effective Size

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$

- [KRT'00] For indentical machines, set $k = m$:

Effective Size

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$

- [KRT'00] For indentical machines, set $k = m$:
  - total β_m summed over all jobs $\geq m \quad \Rightarrow \quad$ expected OPT $= \Omega(1)$.

Effective Size

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$

- [KRT'00] For indentical machines, set $k = m$:
  - total β_m summed over all jobs $\geq m \quad \Rightarrow \quad$ expected OPT $= \Omega(1)$.
  - β_m for jobs on each machine $\leq 1 \quad \Rightarrow \quad$ expected makespan $= O(1)$.

Effective Size

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$

- [KRT'00] For indentical machines, set $k = m$:
  - total β_m summed over all jobs $\geq m \quad \Rightarrow \quad$ expected OPT $= \Omega(1)$.
  - β_m for jobs on each machine $\leq 1 \quad \Rightarrow \quad$ expected makespan $= O(1)$.
    - Pr[one machine load $\geq 1 + c] \leq (1/m)^c$. (next slide.)
    - union bound over all machines.

Effective Size

For any random variable X and parameter $k > 1$, define

$$\textbf{effective size} \quad \beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\left[e^{\log k \cdot X}\right].$$

- [KRT'00] For indentical machines, set $k = m$:
  - total β_m summed over all jobs $\geq m \quad \Rightarrow \quad$ expected OPT $= \Omega(1)$.
  - β_m for jobs on each machine $\leq 1 \quad \Rightarrow \quad$ expected makespan $= O(1)$.
    - Pr[one machine load $\geq 1 + c] \leq (1/m)^c$. (next slide.)
    - union bound over all machines.
  - Hence $O(1)$-approximation.

Lemma (Upper Bound)

For indep. r.v.s $Y_1, \ldots, Y_n$, *if* $\sum_i \beta_k(Y_i) \leq 1$ *then* $\Pr[\sum_i Y_i \geq 1 + c] \leq \frac{1}{k^c}$.

Lemma (Upper Bound)

For indep. r.v.s $Y_1, \ldots, Y_n$, *if* $\sum_i \beta_k(Y_i) \leq 1$ *then* $\Pr[\sum_i Y_i \geq 1 + c] \leq \frac{1}{k^c}$.

$$\Pr[\sum_i Y_i \geq 1 + c] = \Pr[e^{(\log k)\sum_i Y_i} \geq e^{(\log k)(1+c)}] \leq \frac{\mathbb{E}[e^{(\log k)\sum_i Y_i}]}{e^{(\log k)(1+c)}}$$

$$= \frac{\prod_i \mathbb{E}[e^{(\log k) Y_i}]}{e^{(\log k)(1+c)}}$$

Lemma (Upper Bound)

For indep. r.v.s $Y_1, \ldots, Y_n$, *if* $\sum_i \beta_k(Y_i) \leq 1$ *then* $\Pr[\sum_i Y_i \geq 1 + c] \leq \frac{1}{k^c}$.

$$\Pr[\sum_i Y_i \geq 1 + c] = \Pr[e^{(\log k)\sum_i Y_i} \geq e^{(\log k)(1+c)}] \leq \frac{\mathbb{E}[e^{(\log k)\sum_i Y_i}]}{e^{(\log k)(1+c)}}$$

$$= \frac{\prod_i \mathbb{E}[e^{(\log k) Y_i}]}{e^{(\log k)(1+c)}}$$

Taking logarithms,

$$\log \Pr[\sum_i Y_i \geq 1 + c] \leq (\log k) \cdot \left[\sum_i \beta_p(Y_i) - (1 + c)\right] \leq (\log k) \cdot (-c).$$

Effective size for Unrelated Machines Setting

Effective Size: Unrelated Machines

- Must use different k for different kinds of jobs. (Fixed k fails.)

Effective Size: Unrelated Machines

- Must use different k for different kinds of jobs. (Fixed k fails.)

All jobs have size$\sim (0, 1)$ Bernoulli r.v. with $p = \frac{1}{\sqrt{m}}$.

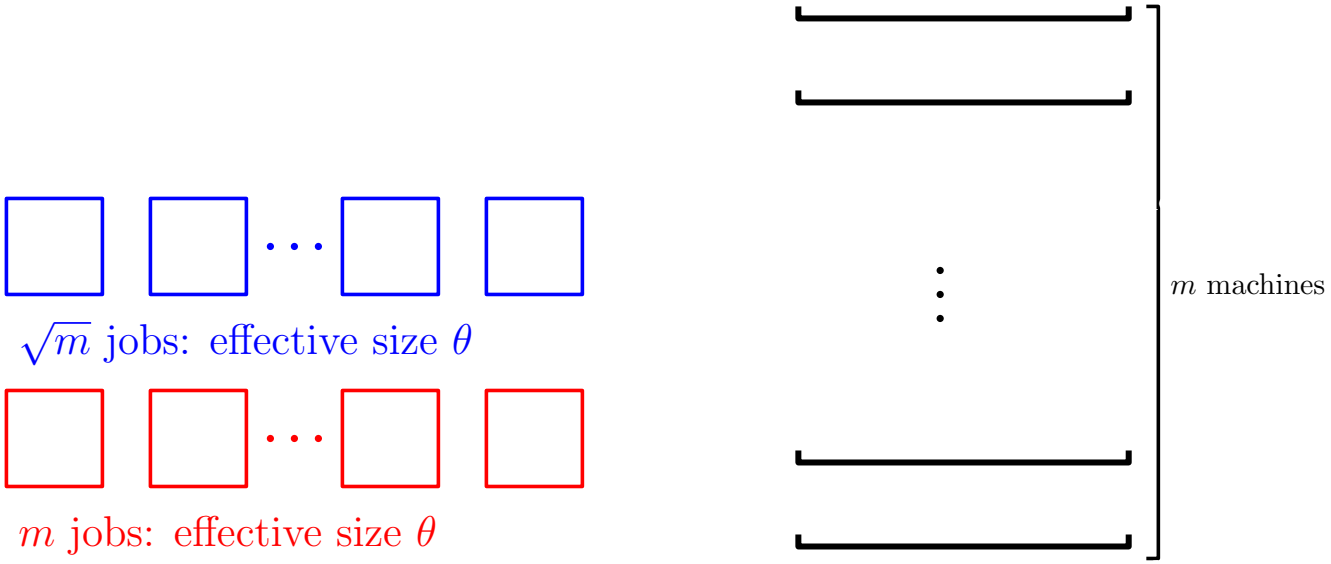
- Type 1: can only be assigned to the first machine.
- Type 2: can be assigned to any machine.

Effective Size: Unrelated Machines

- Must use different k for different kinds of jobs. (Fixed k fails.)

All jobs have size$\sim (0,1)$ Bernoulli r.v. with $p = \frac{1}{\sqrt{m}}$.

- Type 1: can only be assigned to the first machine.
- Type 2: can be assigned to any machine.

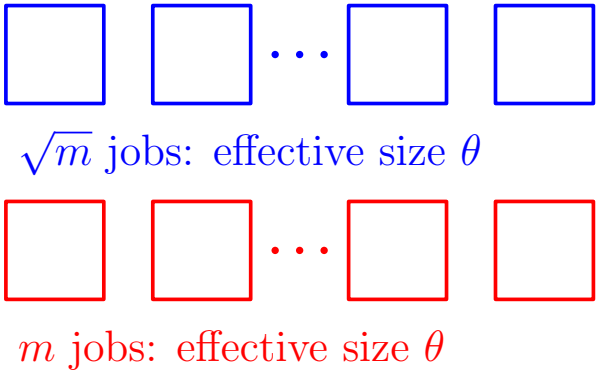


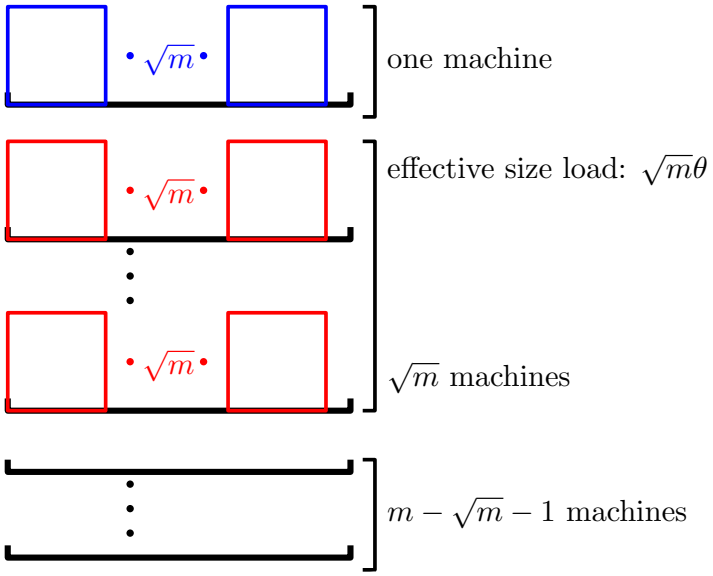
Effective Size: Unrelated Machines

- Must use different k for different kinds of jobs. (Fixed k fails.)

All jobs have size$\sim (0,1)$ Bernoulli r.v. with $p = \frac{1}{\sqrt{m}}$.

- Type 1: can only be assigned to the first machine.
- Type 2: can be assigned to any machine.



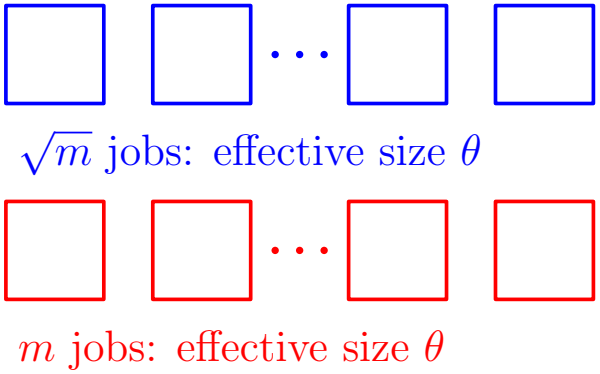


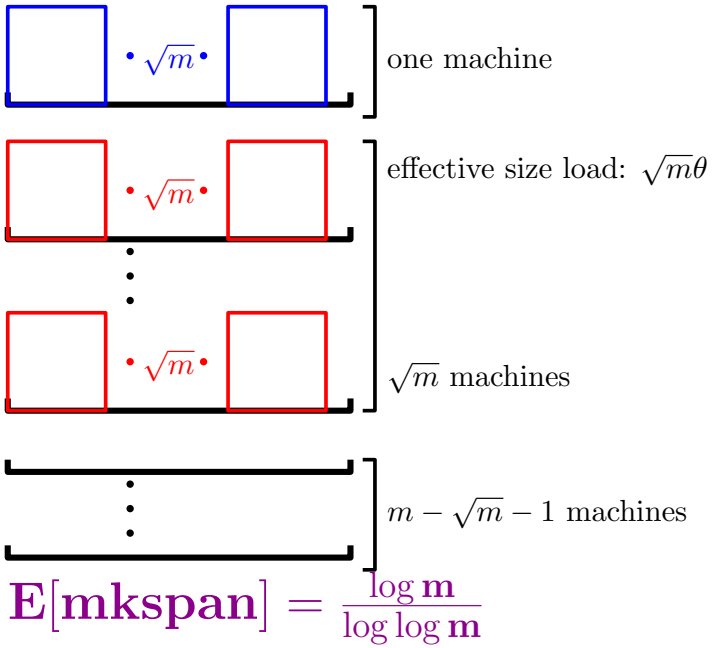
Effective Size: Unrelated Machines

- Must use different k for different kinds of jobs. (Fixed k fails.)

All jobs have size$\sim (0,1)$ Bernoulli r.v. with $p = \frac{1}{\sqrt{m}}$.

- Type 1: can only be assigned to the first machine.
- Type 2: can be assigned to any machine.



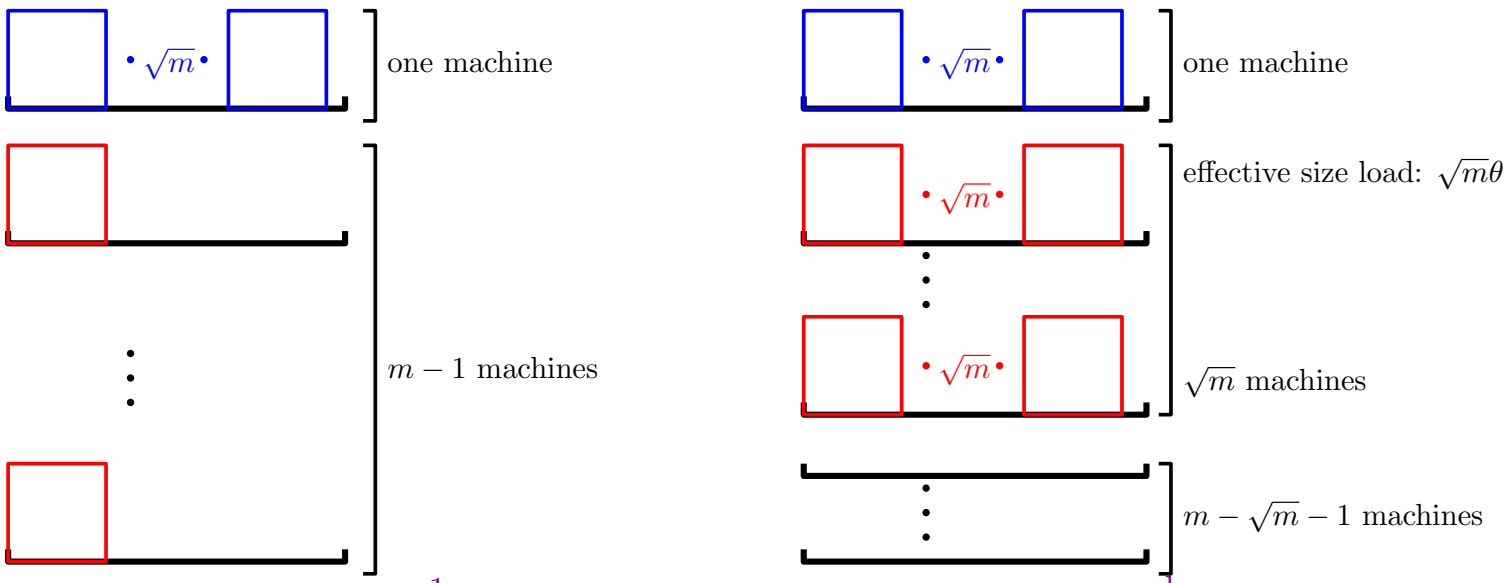


Effective Size: Unrelated Machines

- Must use different k for different kinds of jobs. (Fixed k fails.)

All jobs have size$\sim (0,1)$ Bernoulli r.v. with $p = \frac{1}{\sqrt{m}}$.

- Type 1: can only be assigned to the first machine.
- Type 2: can be assigned to any machine.



$\mathbf{E[mkspan]} = 1 + \frac{1}{e}$ $\qquad$ $\mathbf{E[mkspan]} = \frac{\log \mathbf{m}}{\log\log \mathbf{m}}$

Our Solution for Unrelated Machines

Our Approach

1. valid inequalities using effective size.
2. (large) LP relaxation.
3. Rounding algorithm.

Valid Inequalities

Lemma (New Valid Inequalities)

If assignment satisfies $\mathbb{E}\left[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}\right] \leq 1$, then

$$\sum_{i \in K} \sum_{j \in J_i} \beta_{|K|}(X_{ij}) \;\leq\; O(|K|) \qquad \forall K \subseteq [m].$$

LP Relaxation (cont'd)

- Solve the LP using separation oracle = sorting.

LP Relaxation (cont'd)

- Solve the LP using separation oracle = sorting.
  - LP infeasible $\implies$ optimal expected makespan > 1.
    - Contrapositive to above lemma.
  - LP feasible $\implies$ round fractional solution to satisfy subset of constraints
    - Suffice to bound E[makespan], show next.

LP Relaxation (cont'd)

- Solve the LP using separation oracle = sorting.
  - LP infeasible $\implies$ optimal expected makespan > 1.
    - Contrapositive to above lemma.
  - LP feasible $\implies$ round fractional solution to satisfy subset of constraints
    - Suffice to bound E[makespan], show next.
- Construct instance of unrelated machines problem
- Use rounding from [Lenstra, Shmoys, Tardos 92], [Shmoys, Tardos 93].

Valid Inequalities

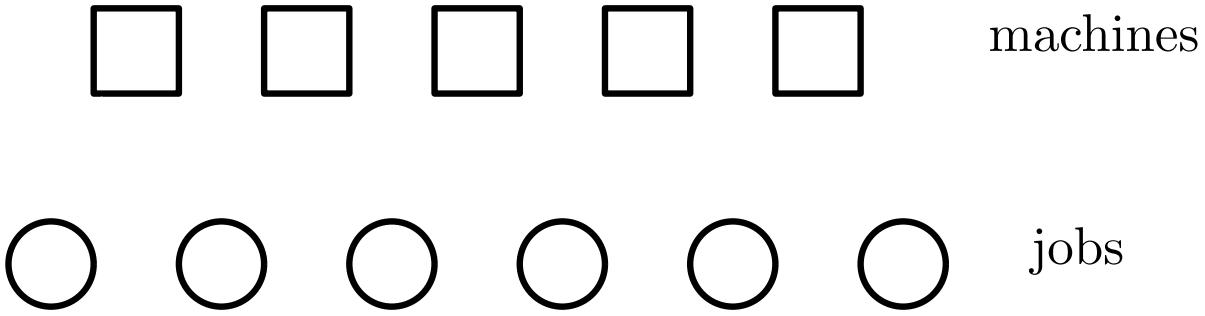
Lemma (New Valid Inequalities)

If assignment satisfies $\mathbb{E}\left[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}\right] \leq 1$, then

$$\sum_{i \in K} \sum_{j \in J_i} \beta_{|K|}(X_{ij}) \;\leq\; O(|K|) \qquad \forall K \subseteq [m].$$

"Pre-Rounding" Algorithm

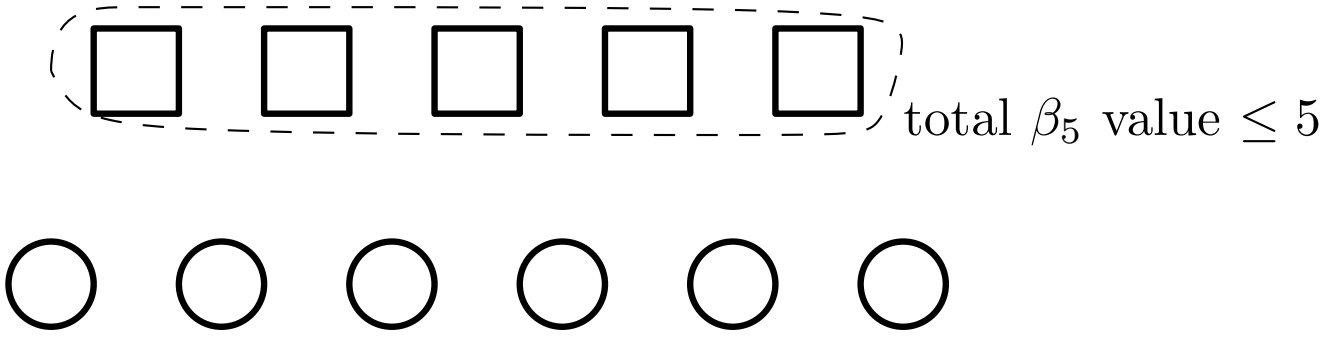
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

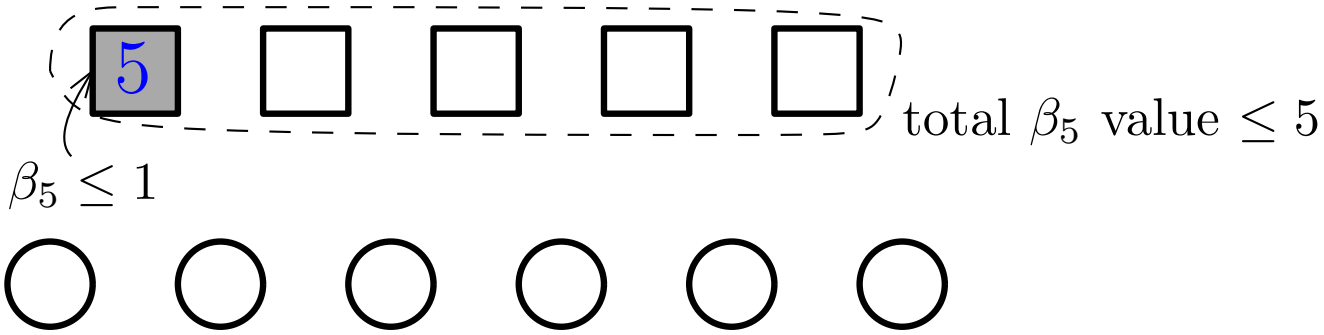
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

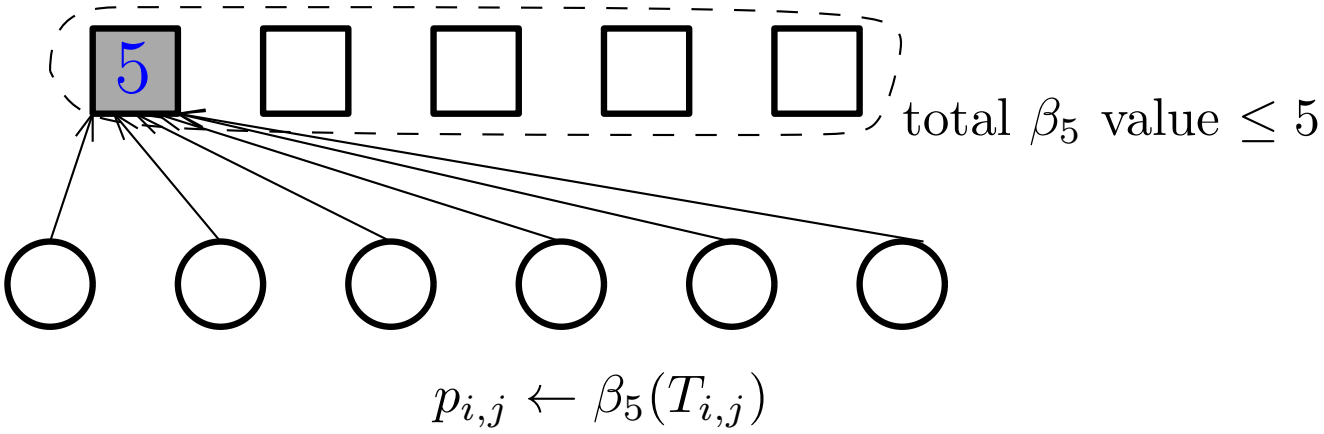
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

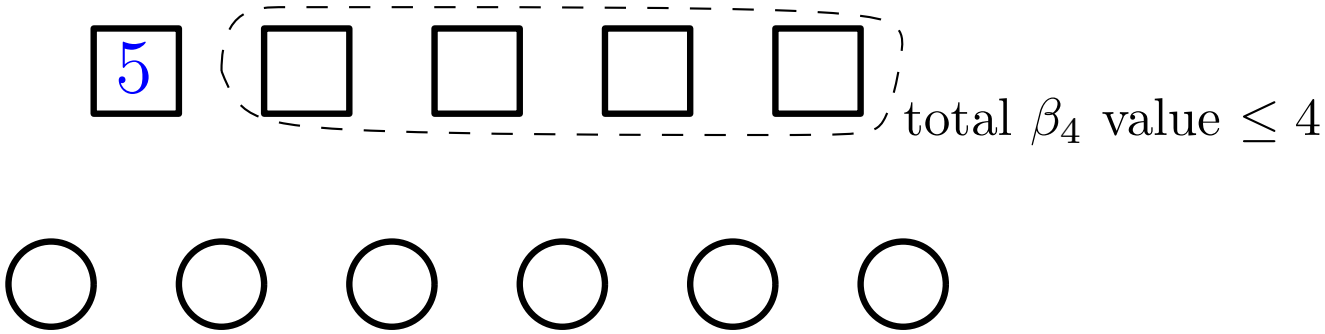
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

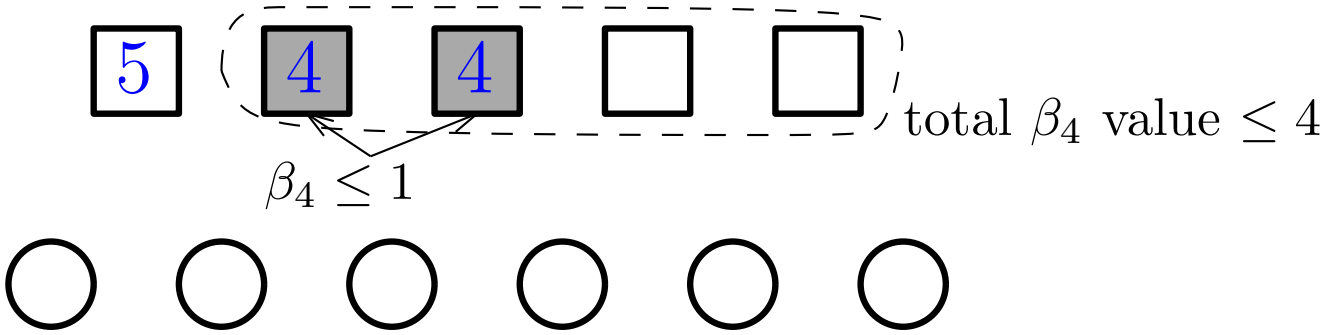
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

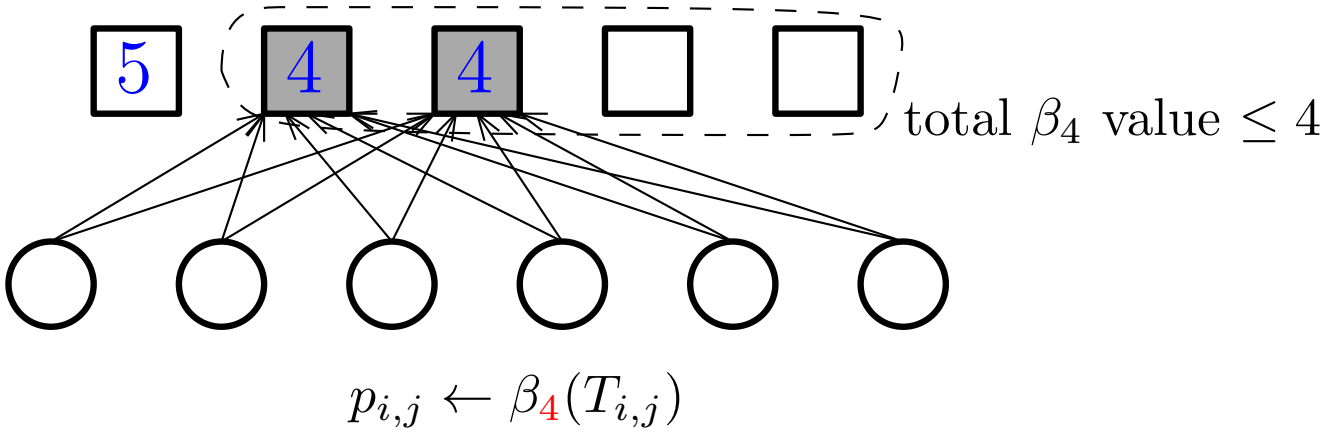
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

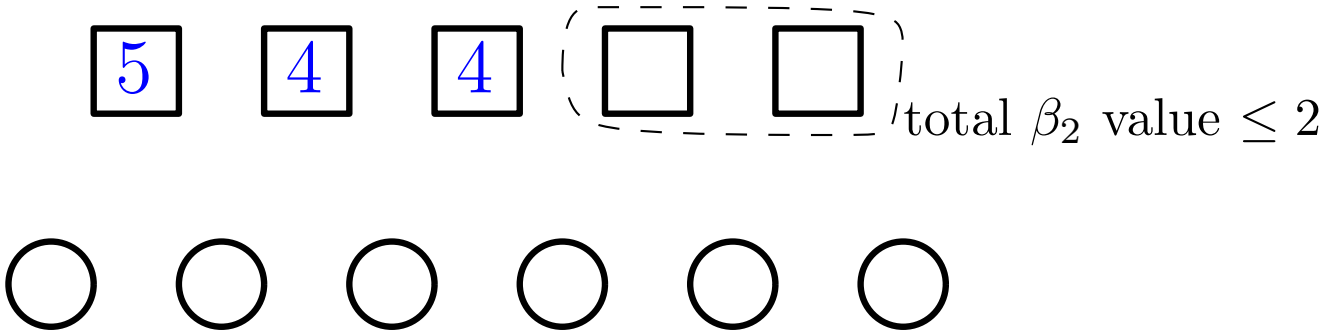
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

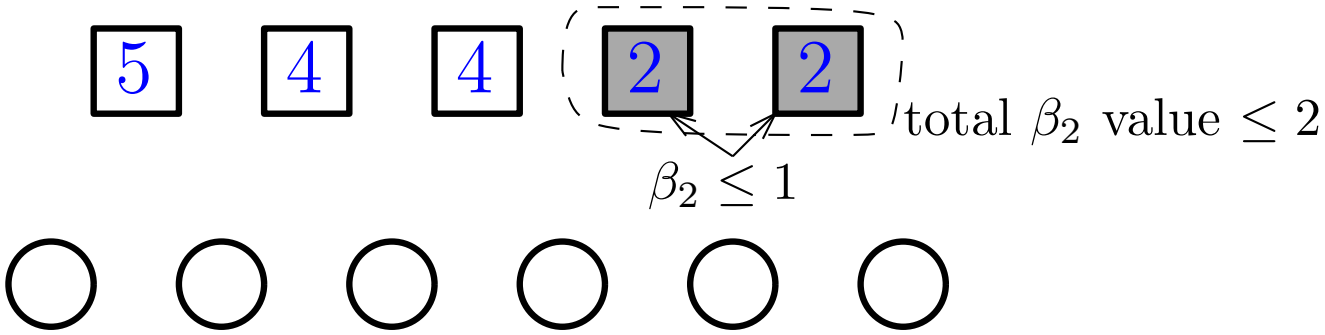
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

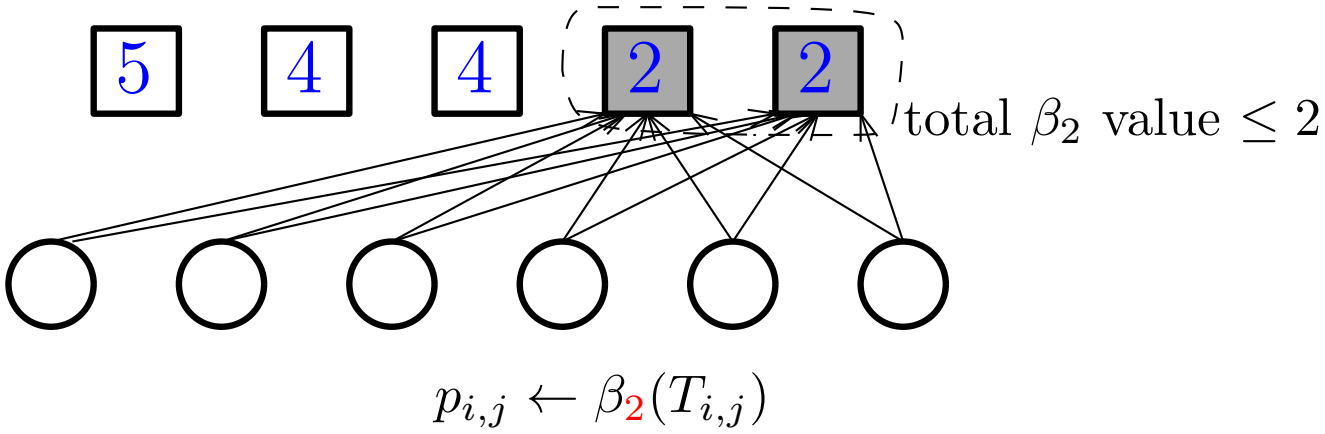
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

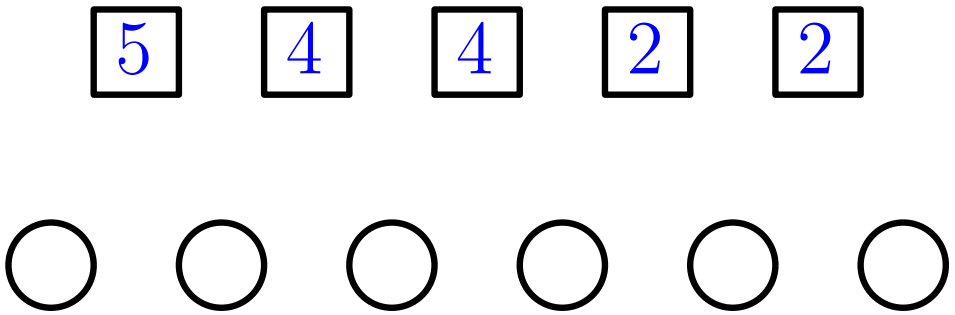
1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

"Pre-Rounding" Algorithm

1. remaining machines $L \leftarrow [m]$, $\ell = |L|$.
2. while $\ell > 0$:
   1. "class ℓ machines: $L' \leftarrow \{i \in L : z_i(\ell) \leq 1\}$.
   2. $p_{ij} \leftarrow \beta_\ell(X_{ij})$ for machines $i \in L'$, all jobs.
   3. $L \leftarrow L \setminus L'$ and $\ell = |L|$.



$$\sum_{i \in K} \beta_{|K|}(X_{ij}) \cdot y_{ij} \leq |K|, \quad \forall K \subseteq [m]$$

Analysis Sketch

- LP gives fractional sol. for unrelated machines (target load $B = 1$).
- Round it: 2-approx rounded assignment $\{J_i\}_{i=1}^m$ satisfies:

$$\sum_{j \in J_i} \beta_{\ell(i)}(X_{ij}) \;\leq\; 2, \quad \forall i \in [m]$$

Analysis Sketch

- LP gives fractional sol. for unrelated machines (target load $B = 1$).
- Round it: 2-approx rounded assignment $\{J_i\}_{i=1}^m$ satisfies:

$$\sum_{j \in J_i} \beta_{\ell(i)}(X_{ij}) \;\leq\; 2, \quad \forall i \in [m]$$

- $\Pr[\text{machine } i \text{ load} \geq 2 + c] \leq \frac{1}{\ell(i)^c}$.

Analysis Sketch

- LP gives fractional sol. for unrelated machines (target load $B = 1$).
- Round it: 2-approx rounded assignment $\{J_i\}_{i=1}^m$ satisfies:

$$\sum_{j \in J_i} \beta_{\ell(i)}(X_{ij}) \;\leq\; 2, \quad \forall i \in [m]$$

- $\Pr[\text{machine } i \text{ load} \geq 2 + c] \leq \frac{1}{\ell(i)^c}$.
- Union bound over machines gives

$$\Pr[\mathit{maxload} > 2 + c] \leq \sum_i \frac{1}{\ell(i)^c} \leq \sum_i \frac{1}{i^c} \ll 1.$$

- Can also bound $\mathbb{E}[\mathit{maxload}]$.

wrapup

- $O(1)$-approx for stochastic load balancing on unrelated machines.
  - Main tool: effective size

wrapup

- $O(1)$-approx for stochastic load balancing on unrelated machines.
  - Main tool: effective size
- Extensions (in the paper):
  - $O(1)$-approx for *budgeted* makespan minimization.
  - $O(\frac{q}{\log q})$-approx for q-norm minimization.
    - Now $O(1)$-approx by M. Molinaro (forthcoming)
- Tighter analysis (2?) even for identical machines?

wrapup

- $O(1)$-approx for stochastic load balancing on unrelated machines.
  - Main tool: effective size
- Extensions (in the paper):
  - $O(1)$-approx for *budgeted* makespan minimization.
  - $O(\frac{q}{\log q})$-approx for q-norm minimization.
    - Now $O(1)$-approx by M. Molinaro (forthcoming)
- Tighter analysis (2?) even for identical machines?
- Incorporate routing aspects?

wrapup

- $O(1)$-approx for stochastic load balancing on unrelated machines.
  - Main tool: effective size
- Extensions (in the paper):
  - $O(1)$-approx for *budgeted* makespan minimization.
  - $O(\frac{q}{\log q})$-approx for q-norm minimization.
    - Now $O(1)$-approx by M. Molinaro (forthcoming)
- Tighter analysis (2?) even for identical machines?
- Incorporate routing aspects?
- Data center issues?

wrapup

- $O(1)$-approx for stochastic load balancing on unrelated machines.
  - Main tool: effective size
- Extensions (in the paper):
  - $O(1)$-approx for *budgeted* makespan minimization.
  - $O(\frac{q}{\log q})$-approx for q-norm minimization.
    - Now $O(1)$-approx by M. Molinaro (forthcoming)
- Tighter analysis (2?) even for identical machines?
- Incorporate routing aspects?
- Data center issues?

Thank you!