Online Unrelated Machine Load Balanceing with Predictions Revisited

Shi Li [1,2] Jiayi Xian[1,2] (presentor)

[1]Computer Sciense Department, University at Buffalo
[2]equal contribution

ICML, July 2021

Outline

1. Introduction
   - Problem settings
     - Unrelated machine restricted assignment setting
     - Identical machine restricted assignment setting
     - Scheduling problem with prediction
   - Known results
2. Techniques
   - Primal Dual
   - Rounding algorithms

Problem setting

Unrelated machine restricted assignment setting

Input: J: jobs

M: machines

$p_{i,j}$: the processing time of job j on machine i

M_j: $\{i \in M : p_{i,j} < \infty\}$ permissible machines for job j

Output: $\sigma : J \mapsto M$: assignments of jobs J on all the machine M

minimize $\max_{i \in M}\{\sum_{j \in \sigma^{-1}(i)} p_{i,j}\}$

Problem setting

Unrelated machine restricted assignment setting

Input: J: jobs
M: machines
$p_{i,j}$: the processing time of job j on machine i
M_j: $\{i \in M : p_{i,j} < \infty\}$ permissible machines for job j

Output: $\sigma : J \mapsto M$: assignments of jobs J on all the machine M
minimize $\max_{i \in M}\{\sum_{j \in \sigma^{-1}(i)} p_{i,j}\}$

- offline setting: $\{p_{i,j}\}$ are given upfront.

Problem setting

Unrelated machine restricted assignment setting

Input: J: jobs
M: machines
$p_{i,j}$: the processing time of job j on machine i
M_j: $\{i \in M : p_{i,j} < \infty\}$ permissible machines for job j

Output: $\sigma : J \mapsto M$: assignments of jobs J on all the machine M
minimize $\max_{i \in M}\{\sum_{j \in \sigma^{-1}(i)} p_{i,j}\}$

- offline setting: $\{p_{i,j}\}$ are given upfront.
- online setting: $\{p_{i,j}\}$ are revealed when job j arrives. The online algorithm is required to irrevocably assign job to a machine upon its arrival.

Problem setting

Identical machine restricted assignment setting (Online)

Input: M: machines J: jobs

$p_{i,j}$: the processing time of job j on machine i. $p_{i,j} \in \{p_j, \infty\}$

Output: $\sigma : J \mapsto M$: assignments of jobs J on all the machine M

minimize $\max_{i \in M} \{\sum_{j \in \sigma^{-1}(i)} p_{i,j}\}$

- Denoted as P$|\mathrm{restricted}$.
- [Azar et al, Aspnes et al.]:
  tight $O(\log m)$-competitive ratio.f

Unrelated machine restricted assignment setting with learned weights

learning augmented online algorithm

Using machine learned predictions to design algorithms for online combinatorial optimization problems.

Unrelated machine restricted assignment setting with learned weights

learning augmented online algorithm

Using machine learned predictions to design algorithms for online combinatorial optimization problems.

Proportional Allocation Scheme of [Agrawal et al] for $\mathrm{P}|\mathrm{restricted}$

- Recall $\mathrm{P}|\mathrm{restricted}$ setting : $p_{i,j} \in \{p_j, \infty\}$
- M_j : $i \in M_j$ iff $p_{i,j} = p_j$

Unrelated machine restricted assignment setting with learned weights

learning augmented online algorithm

Using machine learned predictions to design algorithms for online combinatorial optimization problems.

Proportional Allocation Scheme of [Agrawal et al] for $P|\mathrm{restricted}$

- Recall $P|\mathrm{restricted}$ setting : $p_{i,j} \in \{p_j, \infty\}$
- M_j : $i \in M_j$ iff $p_{i,j} = p_j$
- Given $w \in \mathbb{R}^M_{\geq 0}$, define

$$x_{i,j}^{(w)} = \begin{cases} \frac{w_i}{w(M_j)} & \text{if } i \in M_j \\ 0 & \text{otherwise} \end{cases}$$

Unrelated machine restricted assignment setting with learned weights

learning augmented online algorithm

Using machine learned predictions to design algorithms for online combinatorial optimization problems.

Proportional Allocation Scheme of [Agrawal et al] for $P|\text{restricted}$

- Recall $P|\text{restricted}$ setting : $p_{i,j} \in \{p_j, \infty\}$
- M_j : $i \in M_j$ iff $p_{i,j} = p_j$
- Given $w \in \mathbb{R}^M_{\geq 0}$, define

$$x^{(w)}_{i,j} = \begin{cases} \frac{w_i}{w(M_j)} & \text{if } i \in M_j \\ 0 & \text{otherwise} \end{cases}$$

- [Agrawal et al]: there exists w such that $x^{(w)}$ is $(1+\epsilon)$-approximate solution to LP (Primal).

Known results

Known Results (with learned weights)

- [Agrawal et al, 2018]: $(1+\epsilon)$-approximately optimum to LP (Primal). for P$|\mathrm{restricted}$

Known Results (without learned weights)

[[Azar et al, Aspnes et al.] tight $O(\log m)$-competitive ratio.

Known results

Known Results (with learned weights)

- [Agrawal et al, 2018]: $(1+\epsilon)$-approximately optimum to LP (Primal). for P$|\mathrm{restricted}$
- [Lattanzi et al, 2020] For P$|\mathrm{restricted}$ setting, with some predicted weight vector $w \in \mathbb{R}^M_{\geq 0}$:

	upper bound	lower bound
deterministic		$\Omega\left(\frac{\log m}{\log\log m}\right)$
randomized	$O(\log^3 \log m)$	$\Omega\left(\frac{\log\log m}{\log\log\log m}\right)$

Known Results (without learned weights)

[[Azar et al, Aspnes et al.] tight $O(\log m)$-competitive ratio.

Our results

Our results

- Main Result: For general unrelated machine model, with a predicted dual vector $\beta \in \mathbb{R}^M_{\geq 0}$, and a weight vector $w \in \mathbb{R}^M_{\geq 0}$, online algorithms achieve tight bounds:

	upper bound	lower bound
deterministic	$O\left(\frac{\log m}{\log\log m}\right)$	$\Omega\left(\frac{\log m}{\log\log m}\right)$
randomized	$O\left(\frac{\log\log m}{\log\log\log m}\right)$	$\Omega\left(\frac{\log\log m}{\log\log\log m}\right)$

Our results

- Main Result: For general unrelated machine model, with a predicted dual vector $\beta \in \mathbb{R}^M_{\geq 0}$, and a weight vector $w \in \mathbb{R}^M_{\geq 0}$, online algorithms achieve tight bounds:

	upper bound	lower bound
deterministic	$O\left(\frac{\log m}{\log\log m}\right)$	$\Omega\left(\frac{\log m}{\log\log m}\right)$
randomized	$O\left(\frac{\log\log m}{\log\log\log m}\right)$	$\Omega\left(\frac{\log\log m}{\log\log\log m}\right)$

- Algorithms are **robust**.

Our results

- Main Result: For general unrelated machine model, with a predicted dual vector $\beta \in \mathbb{R}^M_{\geq 0}$, and a weight vector $w \in \mathbb{R}^M_{\geq 0}$, online algorithms achieve tight bounds:

	upper bound	lower bound
deterministic	$O\left(\frac{\log m}{\log\log m}\right)$	$\Omega\left(\frac{\log m}{\log\log m}\right)$
randomized	$O\left(\frac{\log\log m}{\log\log\log m}\right)$	$\Omega\left(\frac{\log\log m}{\log\log\log m}\right)$

- Algorithms are **robust**.
- Prediction (β, w) is **learnable** by seeing a few past instances, under the model of [Lavastida et al.]

Our techniques for main result

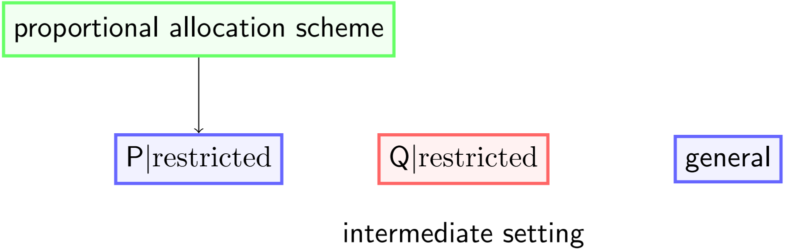
proportional allocation scheme

P|restricted

general

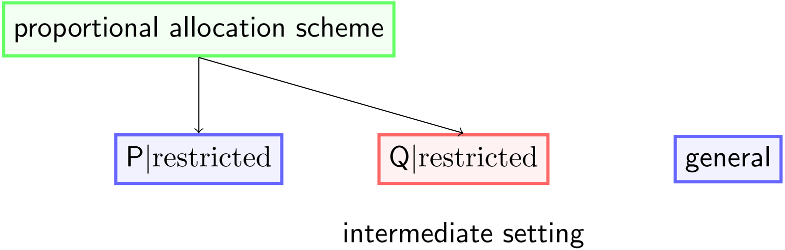
Our techniques for main result

1. We introduce an intermediate setting called related machine restricted assignment setting (Q$|$restricted).



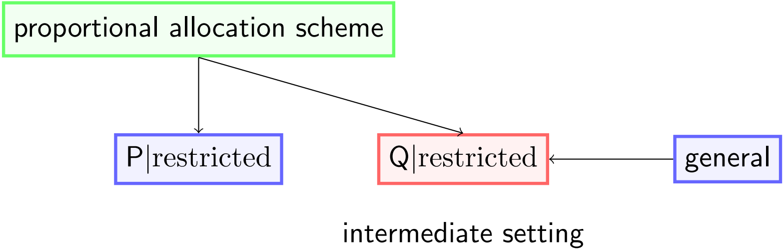
Our techniques for main result

1. We introduce an intermediate setting called related machine restricted assignment setting (Q|restricted).
2. We prove that proportional allocation scheme of [Agrawal et al] also works for Q|restricted setting (easy).



Our techniques for main result

1. We introduce an intermediate setting called related machine restricted assignment setting ($\mathrm{Q}|\mathrm{restricted}$).
2. We prove that proportional allocation scheme of [Agrawal et al] also works for $\mathrm{Q}|\mathrm{restricted}$ setting (easy).
3. We apply Primal-Dual techinque to reduce general setting to $\mathrm{Q}|\mathrm{restricted}$ setting.



Frame Title

We design:

1. deterministic $O\left(\frac{\log m}{\log\log m}\right)$-approximate online rounding algorithm
2. randomized $O\left(\frac{\log\log m}{\log\log\log m}\right)$-approximate online rounding algorithm

Related machine restricted assignment setting

Q|restricted

Input: J: jobs
M: machines
p_j: intrinsic processing time of job j
$s_i \in \mathbb{R}_{>0}$: speed of machine i
$p_{i,j} \in \{\frac{p_j}{s_i}, \infty\}$: the processing time of job j on machine i.

Output: $\sigma : J \mapsto M$: assignments of jobs J on all the machine M
minimize $\max_{i \in M}\{\sum_{j \in \sigma^{-1}(i)} p_{i,j}\}$

Related machine restricted assignment setting

Q|restricted

Input: J: jobs
M: machines
p_j: intrinsic processing time of job j
$s_i \in \mathbb{R}_{>0}$: speed of machine i
$p_{i,j} \in \{\frac{p_j}{s_i}, \infty\}$: the processing time of job j on machine i.

Output: $\sigma : J \mapsto M$: assignments of jobs J on all the machine M
minimize $\max_{i \in M}\{\sum_{j \in \sigma^{-1}(i)} p_{i,j}\}$

- identical machine restricted assignment setting (P|restricted): $p_{i,j} \in \{p_j, \infty\}, \forall i, j$
- related machine restricted assignment setting (Q|restricted): $p_{i,j} \in \left\{\frac{p_j}{s_i}, \infty\right\}, \forall i, j$
- unrelated machine restricted assignment setting (general): $p_{i,j} \in [0, \infty], \forall i, j$

Related machine restricted assignment setting

Q|restricted

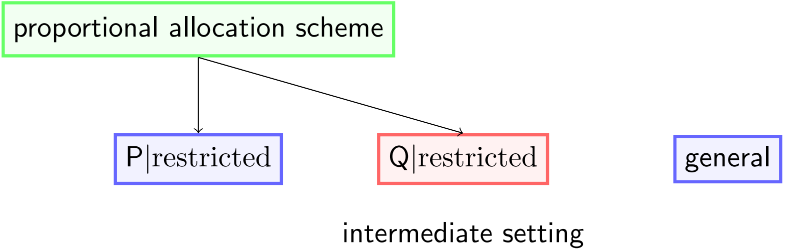
Input: J: jobs
M: machines
p_j: intrinsic processing time of job j
$s_i \in \mathbb{R}_{>0}$: speed of machine i
$p_{i,j} \in \{\frac{p_j}{s_i}, \infty\}$: the processing time of job j on machine i.

Output: $\sigma : J \mapsto M$: assignments of jobs J on all the machine M
minimize $\max_{i \in M}\{\sum_{j \in \sigma^{-1}(i)} p_{i,j}\}$

Lemma

A slight modified version of proportional allocation scheme of [Agrawal et al] works for Q|restricted setting. (easy)

Road map



proportional allocation scheme
P|restricted
Q|restricted
general
intermediate setting

Primal and Dual LPs (unrelated machine model)

$$\min \quad T' \qquad \text{(Primal)}$$

$$\textstyle\sum_{i \in M_j} x_{i,j} = 1 \qquad \forall j \in J \qquad (1)$$

$$\textstyle\sum_{j \in J_i} p_{i,j} x_{i,j} \le T' \qquad \forall i \in M \qquad (2)$$

$$x_{i,j} \ge 0 \qquad \forall i, j \qquad (3)$$

$$\max \quad \textstyle\sum_{j \in J} \alpha_j \qquad \text{(Dual)}$$

$$\alpha_j - p_{i,j}\beta_i \le 0 \qquad \forall i, j \qquad (4)$$

$$\sum_{i \in M} \beta_i = 1 \qquad (5)$$

$$\beta_i \ge 0 \qquad \forall i \in M \qquad (6)$$

Primal and Dual LPs (unrelated machine model)

$$\min \quad T' \qquad \text{(Primal)}$$

$$\textstyle\sum_{i \in M_j} x_{i,j} = 1 \qquad \forall j \in J \qquad (1)$$

$$\textstyle\sum_{j \in J_i} p_{i,j} x_{i,j} \le T' \qquad \forall i \in M \qquad (2)$$

$$x_{i,j} \ge 0 \qquad \forall i, j \qquad (3)$$

$$\max \quad \textstyle\sum_{j \in J} \alpha_j \qquad \text{(Dual)}$$

$$\alpha_j - p_{i,j}\beta_i \le 0 \qquad \forall i, j \qquad (4)$$

$$\sum_{i \in M} \beta_i = 1 \qquad (5)$$

$$\beta_i \ge 0 \qquad \forall i \in M \qquad (6)$$

- β_i: per-unit-time cost of using machine i $\quad(\rightarrow s_i$: speed i)

Primal and Dual LPs (unrelated machine model)

$$\min \quad T' \tag{Primal}$$

$$\sum_{i \in M_j} x_{i,j} = 1 \qquad \forall j \in J \tag{1}$$

$$\sum_{j \in J_i} p_{i,j} x_{i,j} \le T' \qquad \forall i \in M \tag{2}$$

$$x_{i,j} \ge 0 \qquad \forall i, j \tag{3}$$

$$\max \quad \sum_{j \in J} \alpha_j \tag{Dual}$$

$$\alpha_j - p_{i,j}\beta_i \le 0 \qquad \forall i, j \tag{4}$$

$$\sum_{i \in M} \beta_i = 1 \tag{5}$$

$$\beta_i \ge 0 \qquad \forall i \in M \tag{6}$$

- β_i: per-unit-time cost of using machine i $\quad(\to s_i$: speed i)
- $\alpha_j = \min_i p_{i,j}\beta_i$: minimum cost of processing j $\quad(\to p_j)$

Primal and Dual LPs (unrelated machine model)

$$\min \quad T' \qquad \text{(Primal)}$$

$$\sum_{i \in M_j} x_{i,j} = 1 \qquad \forall j \in J \qquad (1)$$

$$\sum_{j \in J_i} p_{i,j} x_{i,j} \le T' \qquad \forall i \in M \qquad (2)$$

$$x_{i,j} \ge 0 \qquad \forall i, j \qquad (3)$$

$$\max \quad \sum_{j \in J} \alpha_j \qquad \text{(Dual)}$$

$$\alpha_j - p_{i,j}\beta_i \le 0 \qquad \forall i, j \qquad (4)$$

$$\sum_{i \in M} \beta_i = 1 \qquad (5)$$

$$\beta_i \ge 0 \qquad \forall i \in M \qquad (6)$$

- β_i: per-unit-time cost of using machine i $\quad(\rightarrow s_i$: speed i)
- $\alpha_j = \min_i p_{i,j}\beta_i$: minimum cost of processing j $\quad(\rightarrow p_j)$
- Due to (5), $\sum_j \alpha_j$ lower bounds the makespan

Proof of Theorem using Dual

$$\max \quad \sum_{j \in J} \alpha_j \tag{Dual}$$

$$\alpha_j - p_{i,j}\beta_i \leq 0 \qquad \forall i, j \tag{7}$$

$$\sum_{i \in M} \beta_i = 1 \tag{8}$$

$$\beta_i \geq 0 \qquad \forall i \in M \tag{9}$$

Main theorem

There is a vector $\beta \in \mathbb{R}^M_{>0}$, given which the general instance is reduced to a Q$|$restricted instance.

- let (α, β) be optimum dual solution

Proof of Theorem using Dual

$$\max \quad \textstyle\sum_{j\in J} \alpha_j \tag{Dual}$$

$$\alpha_j - p_{i,j}\beta_i \leq 0 \qquad \forall i,j \tag{7}$$

$$\sum_{i\in M} \beta_i = 1 \tag{8}$$

$$\beta_i \geq 0 \qquad \forall i \in M \tag{9}$$

Main theorem

There is a vector $\beta \in \mathbb{R}^M_{>0}$, given which the general instance is reduced to a Q$|\mathrm{restricted}$ instance.

- let (α, β) be optimum dual solution
- **complementary slackness**:
  $x_{i,j} > 0 \quad \Rightarrow \quad \alpha_j = \min_i p_{i,j}\beta_i \quad \Rightarrow \quad \alpha_j = p_{i,j}\beta_i \quad \Leftrightarrow \quad p_{i,j} = \frac{\alpha_j}{\beta_i}$.

Proof of Theorem using Dual

$$\max \quad \sum_{j\in J}\alpha_j \tag{Dual}$$

$$\alpha_j - p_{i,j}\beta_i \le 0 \qquad \forall i,j \tag{7}$$

$$\sum_{i\in M}\beta_i = 1 \tag{8}$$

$$\beta_i \ge 0 \qquad \forall i \in M \tag{9}$$

Main theorem

There is a vector $\beta \in \mathbb{R}^M_{>0}$, given which the general instance is reduced to a Q|$\mathrm{restricted}$ instance.

- let (α, β) be optimum dual solution
- **complementary slackness**:
  $x_{i,j} > 0 \quad \Rightarrow \quad \alpha_j = \min_i p_{i,j}\beta_i \quad \Rightarrow \quad \alpha_j = p_{i,j}\beta_i \quad \Leftrightarrow \quad p_{i,j} = \frac{\alpha_j}{\beta_i}$.
- $p_j := \alpha_j$ be size of j, $s_i := \beta_i$ be speed of i.

Proof of Theorem using Dual

$$\max \quad \sum_{j\in J}\alpha_j \qquad \text{(Dual)}$$

$$\alpha_j - p_{i,j}\beta_i \le 0 \qquad \forall i,j \qquad (7)$$

$$\sum_{i\in M}\beta_i = 1 \qquad (8)$$

$$\beta_i \ge 0 \qquad \forall i \in M \qquad (9)$$

Main theorem

There is a vector $\beta \in \mathbb{R}^M_{>0}$, given which the general instance is reduced to a Q$|$restricted instance.

- let (α, β) be optimum dual solution
- **complementary slackness**:
  $x_{i,j} > 0 \quad \Rightarrow \quad \alpha_j = \min_i p_{i,j}\beta_i \quad \Rightarrow \quad \alpha_j = p_{i,j}\beta_i \quad \Leftrightarrow \quad p_{i,j} = \frac{\alpha_j}{\beta_i}$.
- $p_j := \alpha_j$ be size of j, $s_i := \beta_i$ be speed of i.
- $p_{i,j} > \frac{p_j}{s_i} \quad \Rightarrow \quad x_{i,j} = 0 \quad \Rightarrow \quad$ set $p_{i,j} = \infty$.

Proof of Theorem using Dual

$$\max \quad \sum_{j\in J} \alpha_j \qquad \text{(Dual)}$$
$$\alpha_j - p_{i,j}\beta_i \le 0 \qquad \forall i,j \qquad (7)$$
$$\sum_{i\in M} \beta_i = 1 \qquad (8)$$
$$\beta_i \ge 0 \qquad \forall i \in M \qquad (9)$$

Main theorem

There is a vector $\beta \in \mathbb{R}^M_{>0}$, given which the general instance is reduced to a Q$|\mathrm{restricted}$ instance.

- let (α, β) be optimum dual solution
- **complementary slackness**:
  $x_{i,j} > 0 \quad \Rightarrow \quad \alpha_j = \min_i p_{i,j}\beta_i \quad \Rightarrow \quad \alpha_j = p_{i,j}\beta_i \quad \Leftrightarrow \quad p_{i,j} = \frac{\alpha_j}{\beta_i}$.
- $p_j := \alpha_j$ be size of j, $s_i := \beta_i$ be speed of i.
- $p_{i,j} > \frac{p_j}{s_i} \quad \Rightarrow \quad x_{i,j} = 0 \quad \Rightarrow \quad$ set $p_{i,j} = \infty$.
- In practical, α, β could be zero. $p_{i,j} > (1+\epsilon)\frac{p_j}{s_i}$

Deterministic $O\left(\frac{\log m}{\log\log m}\right)$-Approx. Online Rounding

- Independent rounding $\Rightarrow O\left(\frac{\log m}{\log\log m}\right)$-approx.
- Derandomization using conditional expectation leads a deterministic rounding algorithm.

Deterministic $O\left(\frac{\log m}{\log\log m}\right)$-Approx. Online Rounding

- Independent rounding $\Rightarrow O\left(\frac{\log m}{\log\log m}\right)$-approx.
- Derandomization using conditional expectation leads a deterministic rounding algorithm.

Minimize conditional expectation

Suppose we have the expectation of makespan Φ_{t-1} before time t, When job t arrives, we assign it to a machine $i \in M_t$ to minimize the expectation of makespan Φ_t at time t

.

Deterministic $O\left(\frac{\log m}{\log\log m}\right)$-Approx. Online Rounding

- Independent rounding $\Rightarrow O\left(\frac{\log m}{\log\log m}\right)$-approx.
- Derandomization using conditional expectation leads a deterministic rounding algorithm.

Minimize conditional expectation

Suppose we have the expectation of makespan Φ_{t-1} before time t, When job t arrives, we assign it to a machine $i \in M_t$ to minimize the expectation of makespan Φ_t at time t on condition of makespan at time $t-1$.

Randomized $O\left(\frac{\log\log m}{\log\log\log m}\right)$-Approx. Online Rounding

Randomized $O\left(\frac{\log\log m}{\log\log\log m}\right)$-Approx. Online Rounding

- greatly simplified [Lattanzi et al]

Randomized $O\left(\frac{\log\log m}{\log\log\log m}\right)$-Approx. Online Rounding

- greatly simplified [Lattanzi et al]

1. random assignment for **small jobs** $(\sum_{p_{i,j} < \frac{T'}{\log m}} x_{i,j} < \frac{1}{2})$

Randomized $O\left(\frac{\log\log m}{\log\log\log m}\right)$-Approx. Online Rounding

- greatly simplified [Lattanzi et al]

1. random assignment for **small jobs** $(\sum_{p_{i,j}<\frac{T'}{\log m}} x_{i,j} < \frac{1}{2})$
2. attempt to randomly assign **big jobs**, if the load of machine too large, job fails

Randomized $O\left(\frac{\log\log m}{\log\log\log m}\right)$-Approx. Online Rounding

- greatly simplified [Lattanzi et al]

1. random assignment for **small jobs** $(\sum_{p_{i,j}<\frac{T'}{\log m}} x_{i,j} < \frac{1}{2})$
2. attempt to randomly assign **big jobs**, if the load of machine too large, job fails
3. graph induced by failed big jobs have $O(\log^{O(1)} m)$-sized connected components

Randomized $O\left(\frac{\log\log m}{\log\log\log m}\right)$-Approx. Online Rounding

- greatly simplified [Lattanzi et al]

1. random assignment for **small jobs** ($\sum_{p_{i,j} < \frac{T'}{\log m}} x_{i,j} < \frac{1}{2}$)
2. attempt to randomly assign **big jobs**, if the load of machine too large, job fails
3. graph induced by failed big jobs have $O(\log^{O(1)} m)$-sized connected components
4. using deterministic rounding algorithm for failed jobs

Thank you for your time.