On the Configuration-LP

of the Restricted Assignment Problem

Klaus Jansen, Lars Rohwedder (SODA 2017)

University of Kiel, Germany

Presented By: Hamed Saleh

Definition

Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

Each job has a processing time p_{ij},

that depends on the machine it is assigned to

Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

A solution is a function $\sigma : \mathcal{J} \rightarrow \mathcal{M}$

Assignment Problem

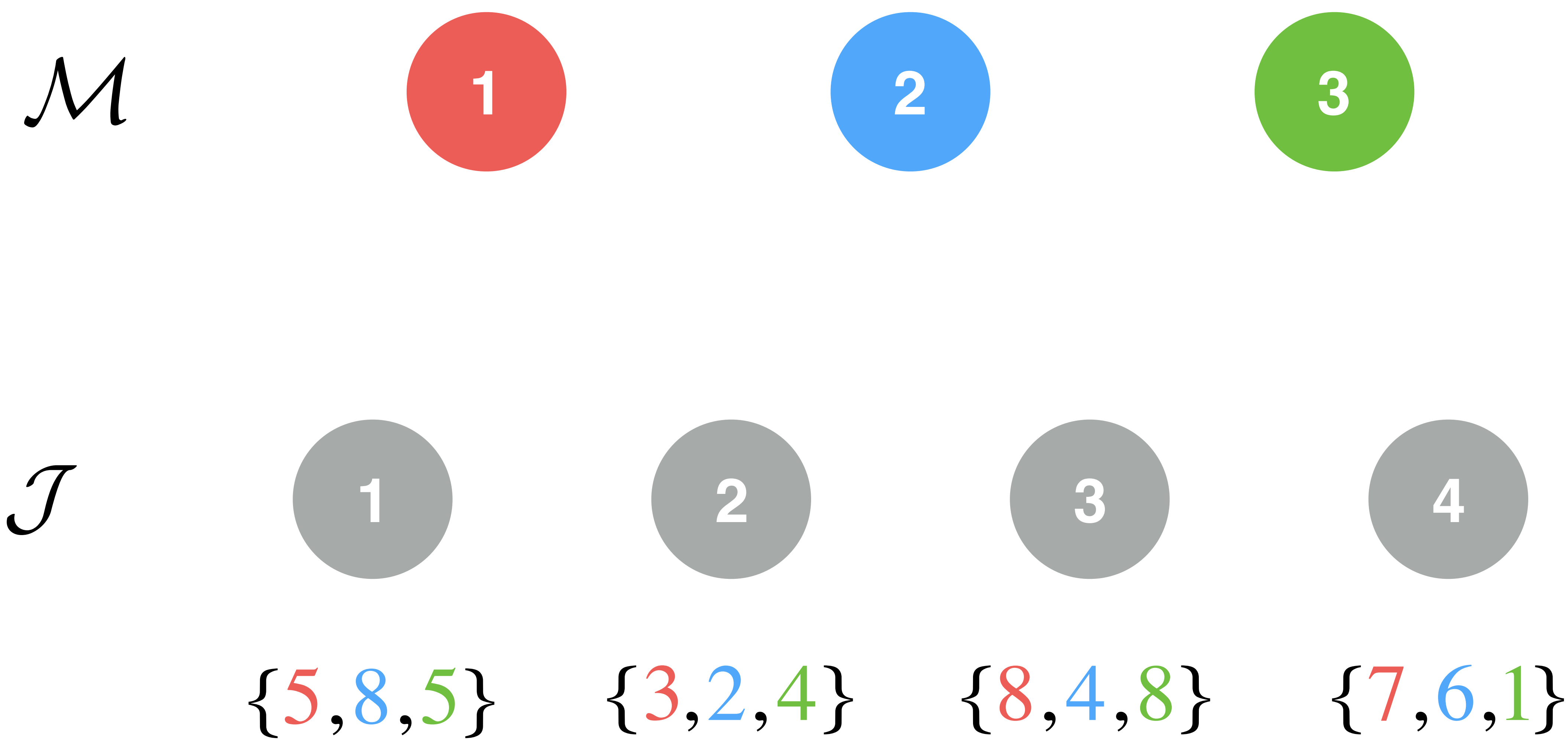
How to distribute jobs among a set of machines

to minimize the **makespan**

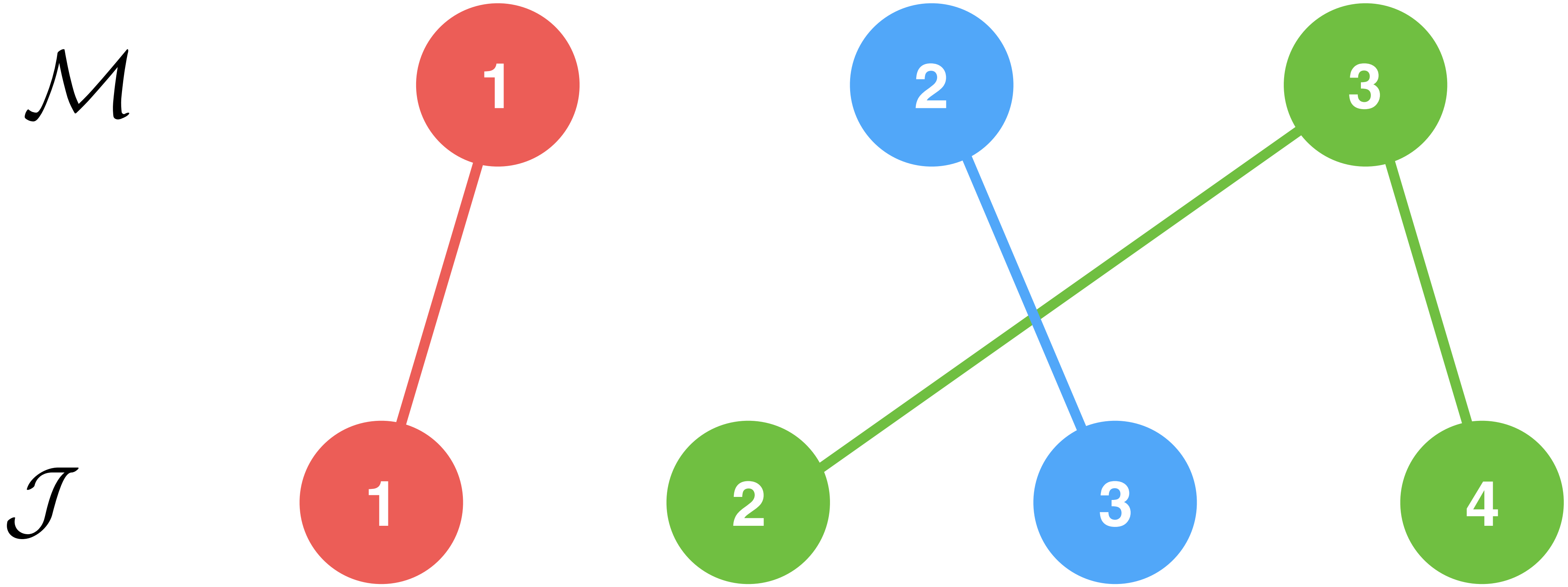
The highest load among all machines

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

Assignment Problem

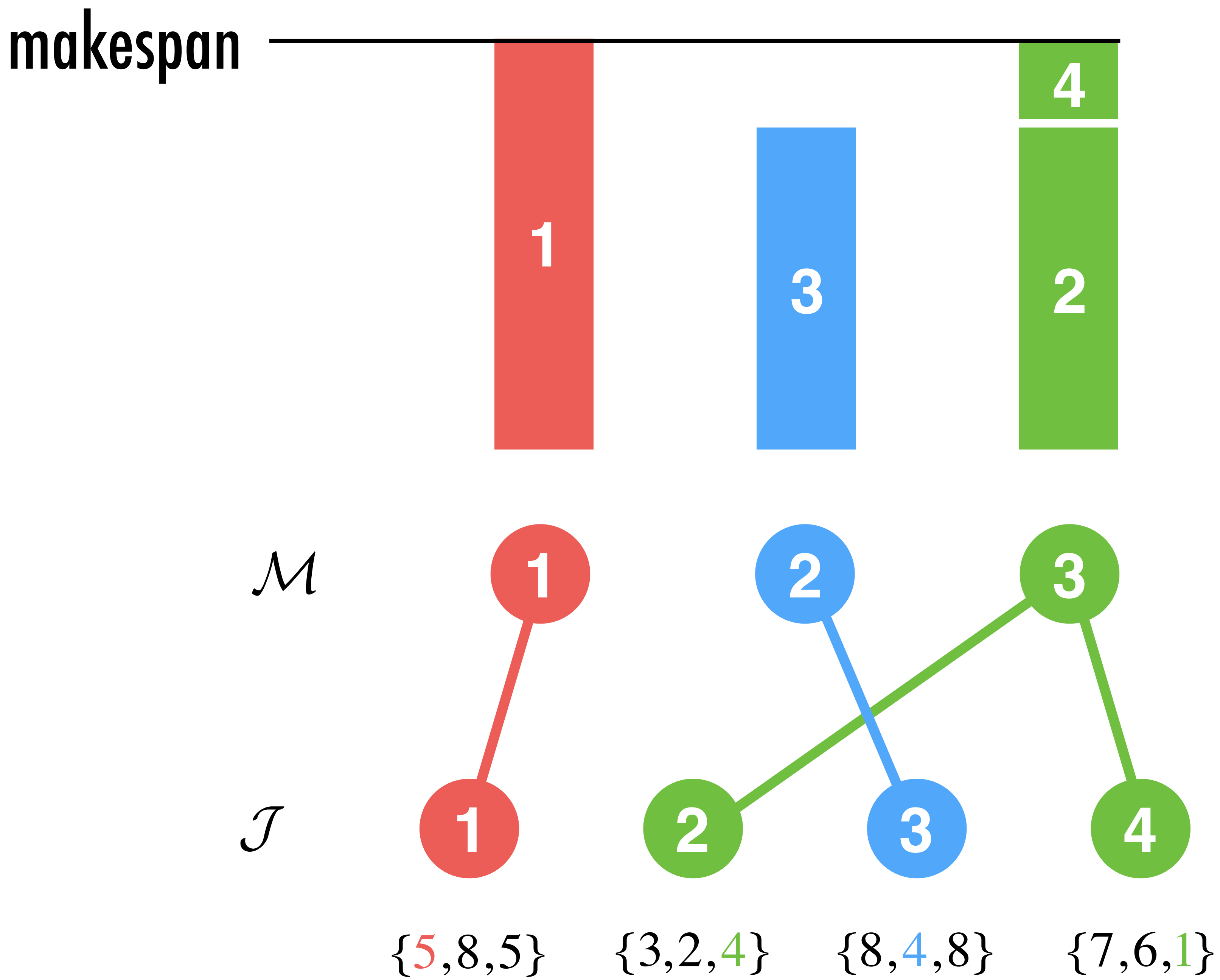


Assignment Problem



$\{5, 8, 5\}$ $\{3, 2, 4\}$ $\{8, 4, 8\}$ $\{7, 6, 1\}$

Assignment Problem



Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

Lenstra, Shmoys, Tardos (1990)

1. gave an approximation guarantee of 2
2. there is no approximation better than $3/2$

Restricted Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

$$p_{ij} \in \{p_j, \infty\}$$

Restricted Assignment Problem

How to distribute jobs among a set of machines

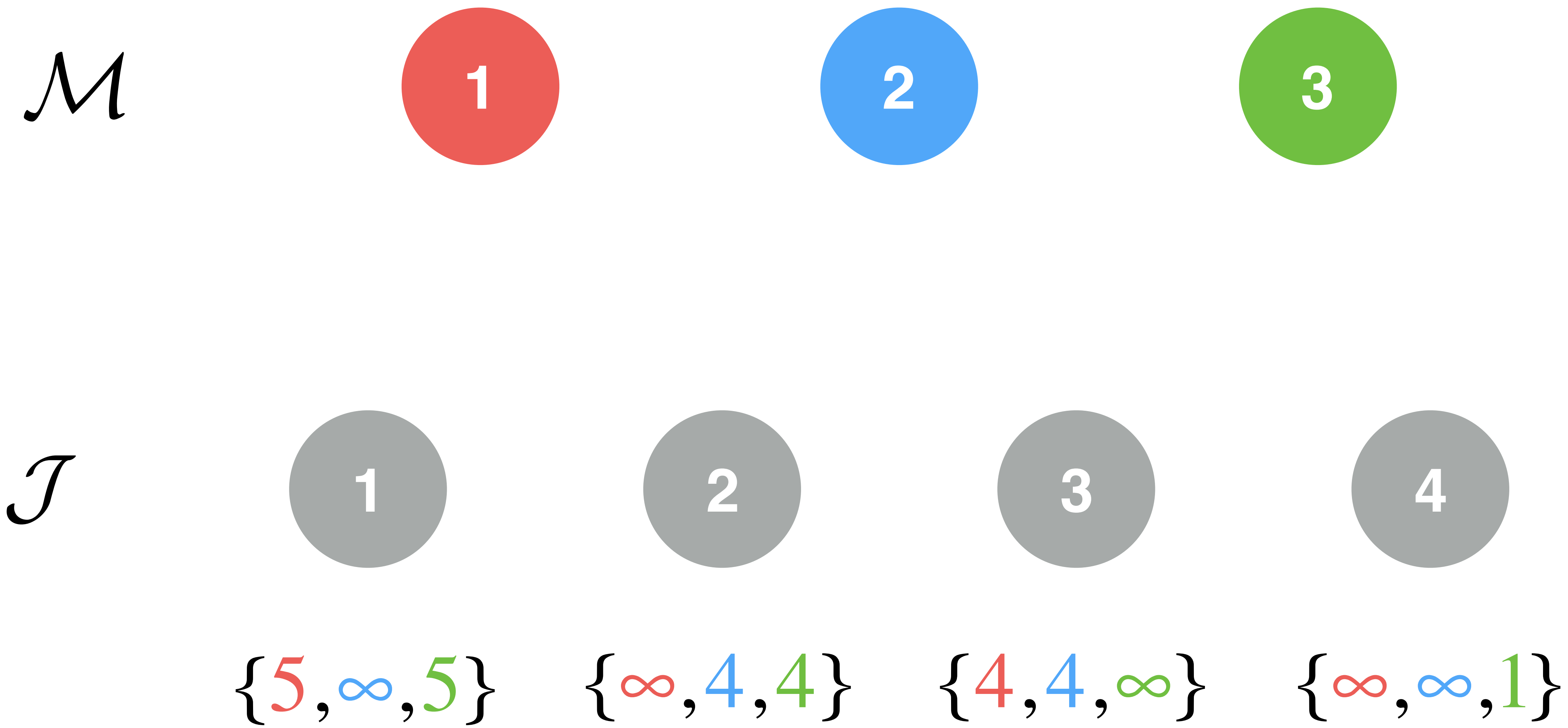
to minimize the **makespan**

$$p_{ij} \in \{p_j, \infty\}$$

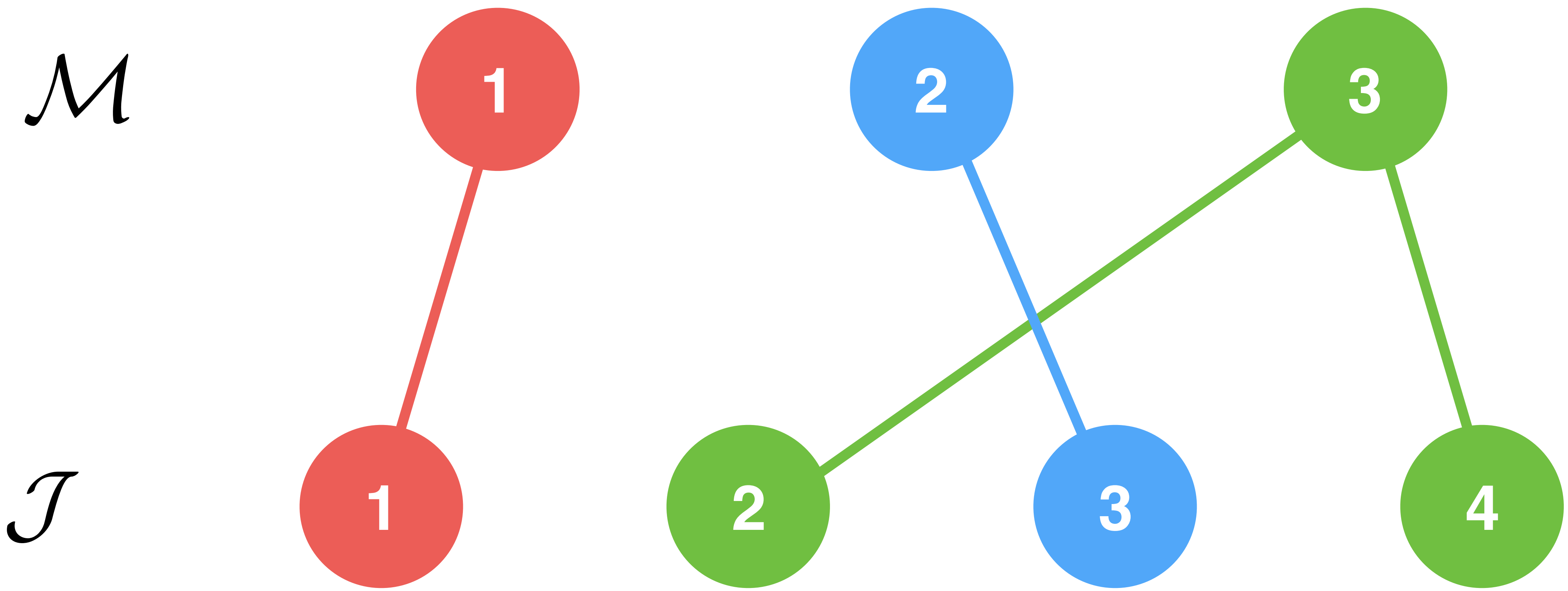
Each job is allowed only on a **subset** of machines,

but the processing time on any of them is **identical**.

Restricted Assignment Problem

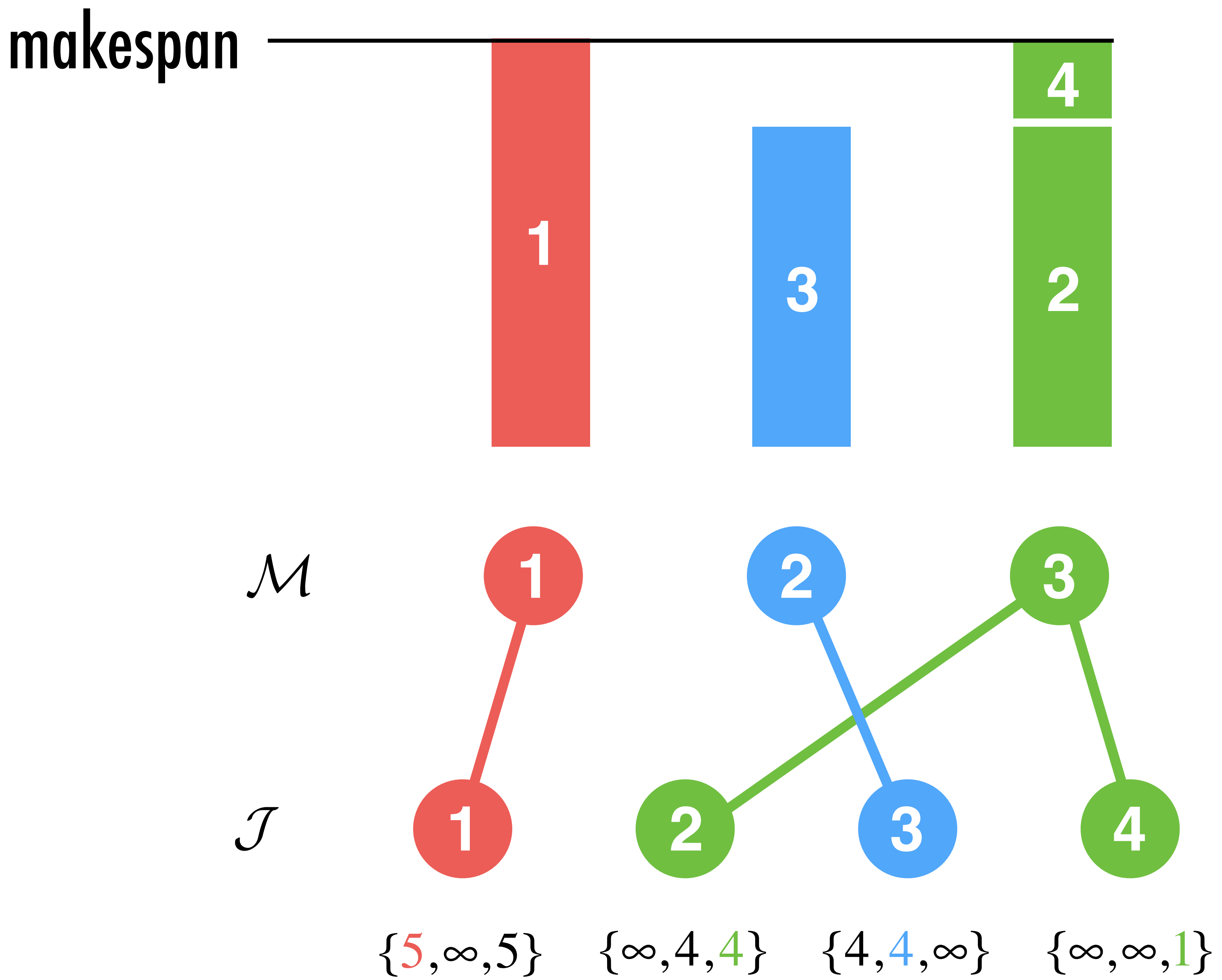


Restricted Assignment Problem



$\{5, \infty, 5\}$ $\{\infty, 4, 4\}$ $\{4, 4, \infty\}$ $\{\infty, \infty, 1\}$

Restricted Assignment Problem



Restricted Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

$$p_{ij} \in \{p_j, \infty\}$$

Svensson (2012)

1. gave an estimation guarantee of $2 - 1/17$

Restricted Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

$$p_{ij} \in \{p_j, \infty\}$$

This paper Jansen, Rohwedder (2016)

1. gave an estimation guarantee of $2 - 1/6$ + simpler approach

Restricted Assignment Problem

How to distribute jobs among a set of machines

to minimize the **makespan**

$$p_{ij} \in \{p_j, \infty\}$$

No approximation guarantee better than 2 so far

Approximation vs. Estimation

What's the deal?

Approximation vs. Estimation

What's the deal?

1. A *c-estimation* algorithm computes a value E with

$$OPT < E < c.OPT$$

2. A *c-approximation* algorithm provides a solution too

Approximation vs. Estimation

What's the deal?

Feige, Jozpeh (2015)

1. There are NP optimization problems for which estimation is easier than approximation

- Unless P = NP or TFNP = FP

Estimating Restricted Assignment Problem

Assignment LP

The most natural linear programming

$$
\begin{aligned}
\sum_{i=1}^{m} y_{ij} &= 1, && j = 1, \ldots, n, \\
\sum_{j=1}^{n} p_{ij} y_{ij} &\le T, && i = 1, \ldots, m, \\
y_{ij} &\ge 0, && \forall i, j.
\end{aligned}
$$

Assignment LP

The most natural linear programming

Integral solution = (Restricted) Assignment Problem solution

Integrality gap: difference between integral and fractional solution

$$y_{ij} \geq 0, \qquad \forall i, j.$$

Configuration LP

A linear programming system with nice integrality gap

in the Restricted Assignment Problem

Configuration LP

A **configuration** for a machine is a set of jobs that would not exceed the target makespan T.

$$\mathcal{C}_i(T) = \{C \subseteq \mathcal{J} : \sum_{j \in C} p_{ij} \leq T\}$$

Configuration LP

The linear programming system is as follows:

$$
\begin{aligned}
\sum_{C \in \mathcal{C}_i(T)} x_{i,C} &\leq 1 && \forall i \in \mathcal{M} \\
\sum_{i \in \mathcal{M}} \sum_{C \in \mathcal{C}_i(T) : j \in C} x_{i,C} &\geq 1 && \forall j \in \mathcal{J} \\
x_{i,C} &\geq 0
\end{aligned}
$$

Primal of the Configuration-LP

Configuration LP

A mixture of configurations is assigned to a machine,

where $x_{i,C}$ shows which fraction of the configuration is assigned.

$$\sum_{C \in \mathcal{C}_i(T)} x_{i,C} \leq 1 \qquad \forall i \in \mathcal{M}$$

$$\sum_{i \in \mathcal{M}} \sum_{C \in \mathcal{C}_i(T): j \in C} x_{i,C} \geq 1 \qquad \forall j \in \mathcal{J}$$

$$x_{i,C} \geq 0$$

Primal of the Configuration-LP

Configuration LP

But the Configuration-LP has an **exponential** number of variables!

$$\sum_{C \in \mathcal{C}_i(T)} x_{i,C} \leq 1 \qquad \forall i \in \mathcal{M}$$

$$\sum_{i \in \mathcal{M}} \sum_{C \in \mathcal{C}_i(T): j \in C} x_{i,C} \geq 1 \qquad \forall j \in \mathcal{J}$$

$$x_{i,C} \geq 0$$

Primal of the Configuration-LP

Configuration LP

But the dual of Configuration-LP has an **exponential** number of constraints!

$$\min \sum_{i \in \mathcal{M}} y_i - \sum_{j \in \mathcal{J}} z_j$$

$$s.t.$$

$$y_i \geq \sum_{j \in C} z_j \quad \forall i \in \mathcal{M}, C \in \mathcal{C}_i(T)$$

$$y_i, z_j \geq 0$$

Dual of the Configuration-LP

Configuration LP

Could be solved using **ellipsoid** method to any desired accuracy.

$$
\begin{aligned}
&\min \sum_{i\in\mathcal{M}} y_i - \sum_{j\in\mathcal{J}} z_j \\
&s.t. \\
&\quad y_i \geq \sum_{j\in C} z_j \quad \forall i \in \mathcal{M}, C \in \mathcal{C}_i(T) \\
&y_i, z_j \geq 0
\end{aligned}
$$

Dual of the Configuration-LP

Configuration LP

Theorem 1.1.

The configuration-LP for the Restricted Assignment problem has an integrality gap of at most 2 - 1/6.

- One can estimate Restricted Assignment problem by a factor less than 2 in polynomial time.

Configuration LP

Theorem 1.1.

The configuration-LP for the Restricted Assignment problem has an integrality gap of at most 2 - 1/6.

Jansen, Land, Maack (2016)

1. gave an instance with integrality gap $3/2$

Proof

Local Search Algorithm

Local search algorithm produces a solution with makespan $(1+5/6)OPT^*$.

OPT* = The minimal T for which the LP is feasible.

(From now on, divide everything by OPT*)

Local Search Algorithm

Iterative approach: Inserting jobs one by one, maintaining the makespan condition meanwhile.

Local Search Algorithm

Small and Big jobs: Jobs are divided into two parts. A job is *small* if $p_j \leq 1/2$ and *big* otherwise.

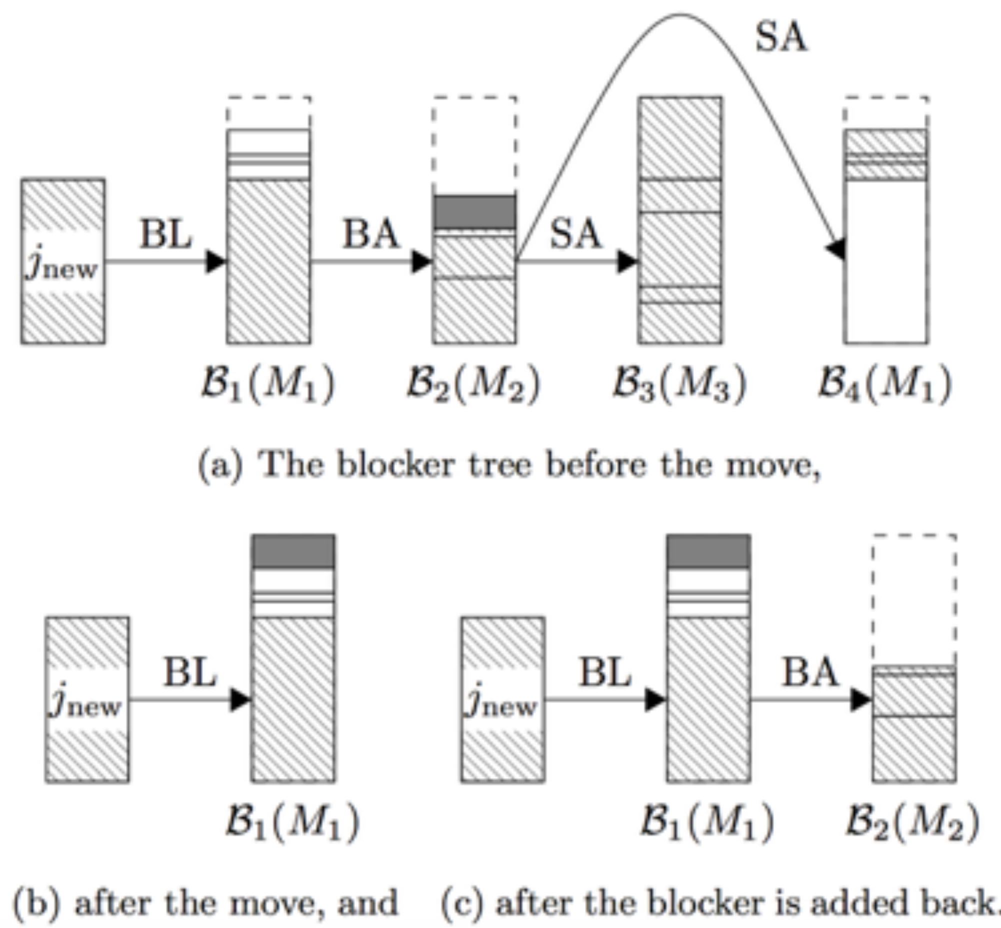
Local Search Algorithm

Blockers: A blocker is a tuple (i, j, θ) which shows moving job j to machine i with type θ.

$$\theta \in \{SA, BA, BB, BL\}$$

Local Search Algorithm

The Blocker Tree: A set of blockers $B_1, B_2, \ldots, B_l$, which form a tree structure.



(a) The blocker tree before the move,

(b) after the move, and (c) after the blocker is added back.

Local Search Algorithm

```
// Input: Job j_new and partial schedule σ
initialize empty blocker tree 𝒯;
loop
  if a move (j, i) in 𝒯 is valid then
    Let ℬ_k be the blocker that activated j;
    // Update the schedule
    σ(j) ← i;
    if j = j_new then
      return σ;
    end
    // Delete ℬ_k, ℬ_{k+1}, ...
    𝒯 ← 𝒯^(≤k−1);
  else
    choose a potential move (j,i)
        with minimum value;
    add (j,i) to 𝒯 with the correct type;
  end
end
```

Algorithm does not get stuck

Supposing the algorithm get stuck at some point,

they have shown that the dual LP is **unbounded**.

Therefore, the primal LP is **infeasible**,

which is contradiction.

due to characteristics of configuration-LP

Algorithm terminates

The signature vector is defined as

$$(val(B_1), val(B_2), \ldots, val(B_l), \infty)$$

where the value of l is at most $|\mathcal{J}| \times |\mathcal{M}|$

Algorithm terminates

The signature vector is defined as

$$(val(B_1), val(B_2), \ldots, val(B_l), \infty)$$

and the signature vector is decreasing lexicographically, thereby terminating eventually.

Proof

Since the algorithm does not get **stuck** at any moment,

and the algorithm **terminates** eventually,

Theorem1.1 is proved. ■

Conclusion

Open Directions

1. Improving the approximation bound in either Restricted or General version of the Assignment Problem.

   Since efficient variants of respective local search algorithm were discovered for Restricted Max-Min Fair Allocation.

Open Directions

2. The estimation bound found in this paper is not tight.

Proposed approach:

A restricted situation where for every job either $p_j = 1$ or $p_j \leq 1/2$

Thank you!