Strong LP formulations for scheduling splittable jobs on unrelated machines

J. Correa, A. Marchetti-Spaccamela, J. Matuschke,
L. Stougie, O. Svensson, V. Verdugo, **J. Verschae**

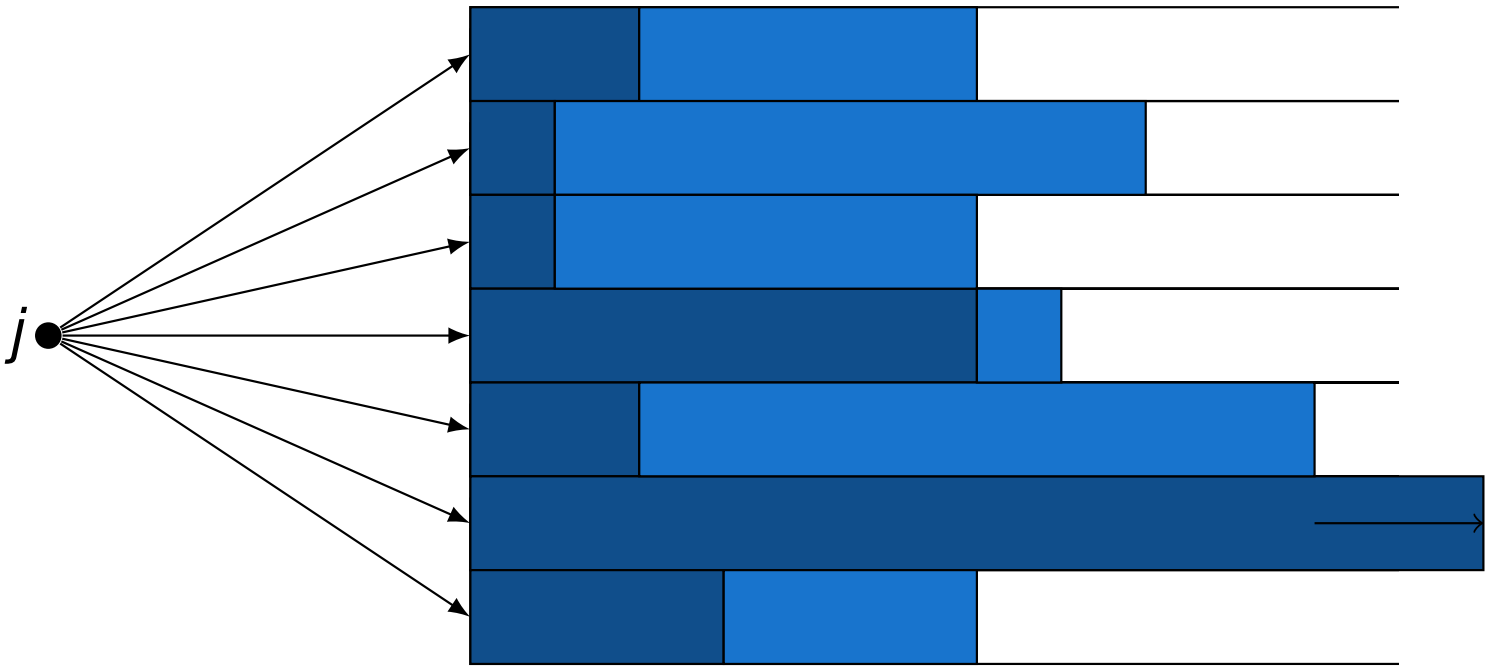
New Challenges in Scheduling Theory

April 1, 2014

Problem definition

Input

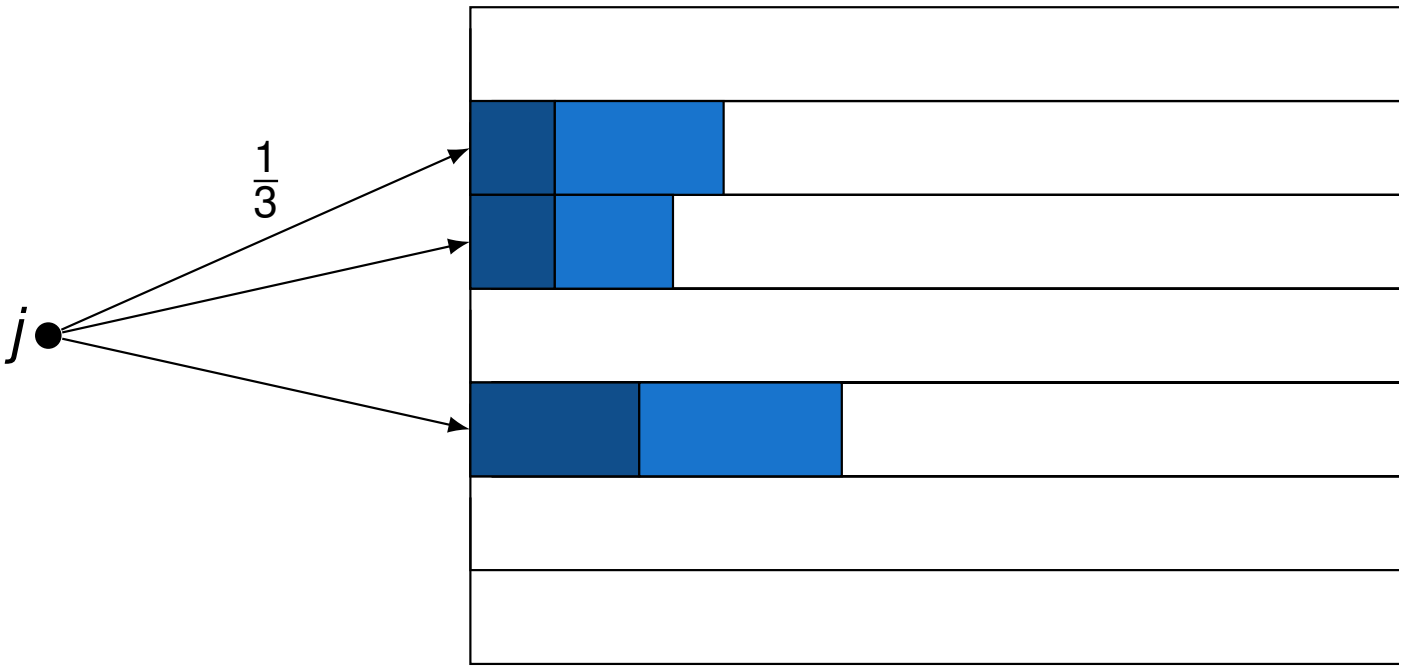
- m unrelated machines M.
- n splittable jobs J,
  - p_{ij}: processing time $j \in J, i \in M$,
  - s_{ij}: setup time $j \in J, i \in M$.



Problem definition

Input

- m unrelated machines M.
- n splittable jobs J,
  - p_{ij}: processing time $j \in J, i \in M$,
  - s_{ij}: setup time $j \in J, i \in M$.



More formally...

x_{ij}: fraction of j assigned to i.

Problem

$$\sum_{i \in M} x_{ij} = 1 \qquad \text{for all } j \in J,$$

$$x_{ij} \geq 0 \qquad \text{for all } j \in J, i \in M,$$

More formally...

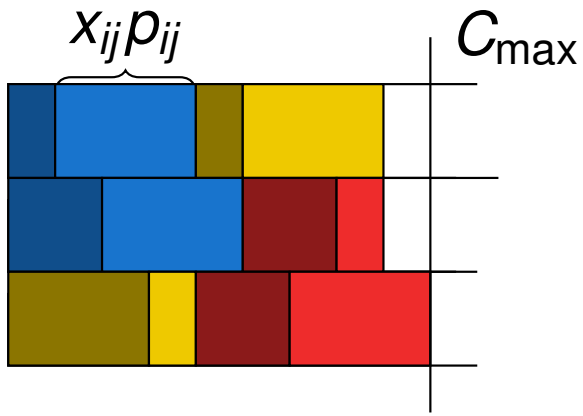
x_{ij}: fraction of j assigned to i.

Problem

$$\sum_{i \in M} x_{ij} = 1 \qquad \text{for all } j \in J,$$

$$x_{ij} \geq 0 \qquad \text{for all } j \in J, i \in M,$$

Minimize: $$C_{\max} = \max_{i \in M} \left\{ \sum_{j : x_{ij} > 0} x_{ij} p_{ij} + s_{ij} \right\}$$



Remarks

Remark 1

Jobs can be run in parallel!

Remarks

Remark 1

Jobs can be run in parallel!

Remark 2

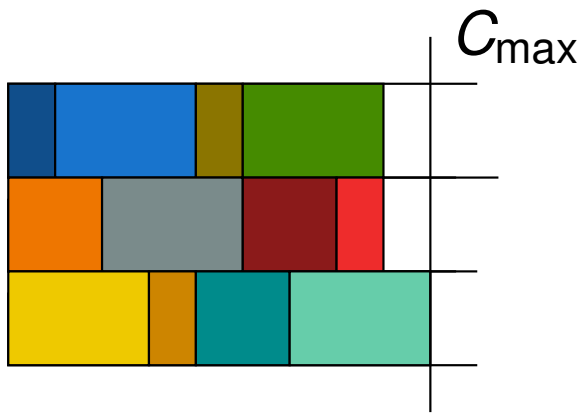
If $s_{ij} = 0$ for all i, j: problem is an LP

$$\begin{aligned} \min \quad & C_{\max} & \\ \sum_{i \in M} x_{ij} &= 1 & \text{for all } j, \\ \sum_{j \in J} x_{ij} p_{ij} &\leq C_{\max} & \text{for all } i, \\ x_{ij} &\geq 0 & . \end{aligned}$$

Remarks

Remark 3

If $p_{ij} = 0$: problem is equivalent to $R||C_{max}$.



Related Work

Classic models

- For $R||C_{max}$: [Lenstra, Shmoys & Tardos]
  - NP-hard to approximate within $3/2 - \varepsilon$,
  - There exists a 2-approx.
- For $R|pmtn|C_{max}$: [Lawler & Labetoulle]
  - Poly-time solvable.

Related Work

Classic models

- For $R||C_{max}$: [Lenstra, Shmoys & Tardos]
  - NP-hard to approximate within $3/2 - \varepsilon$,
  - There exists a 2-approx.
- For $R|pmtn|C_{max}$: [Lawler & Labetoulle]
  - Poly-time solvable.

Split jobs

- For C_{max} on parallel machines: [Chen et al.]
  - There exists a $5/3$-approx.
- For $\sum_j w_j C_j$:
  - 2.781-approx for $p_j = p$ and $w_j = w$. [Schalekamp et al.]
  - 4-approx. for parallel machines. [Correa, Verdugo, V.]
  - $O(1)$-approx. for unrelated machines.

Main results

Theorem

There exists an approximation algorithm with a guarantee of $1 + \phi \approx 2.618$.

Theorem

The problem is hard to approximate within $\frac{e}{e-1} \approx 1.581$.

An LP-relaxation

- Guess $C_{\max} \in [1, \sum_{i,j} p_{ij} + s_{ij}]$:

[LST]-LP

$$\sum_{i \in M} x_{ij} = 1 \qquad \text{for all } j,$$

$$\sum_{j \in J} x_{ij}(p_{ij} + s_{ij}) \leq C_{\max} \qquad \text{for all } i,$$

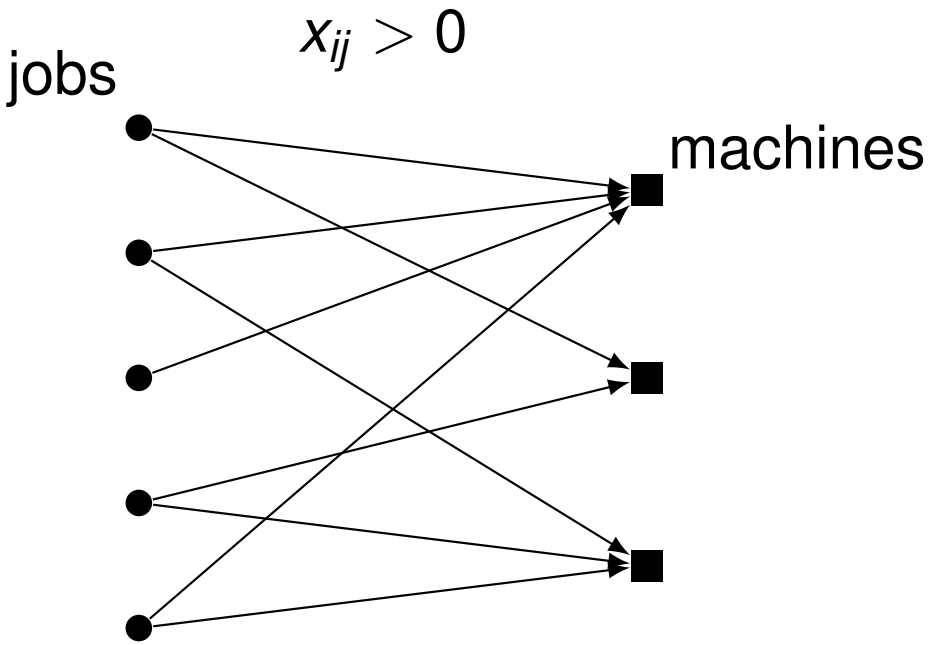
$$x_{ij} = 0 \qquad \text{if: } s_{ij} > C_{\max},$$

$$x_{ij} \geq 0.$$

Definition

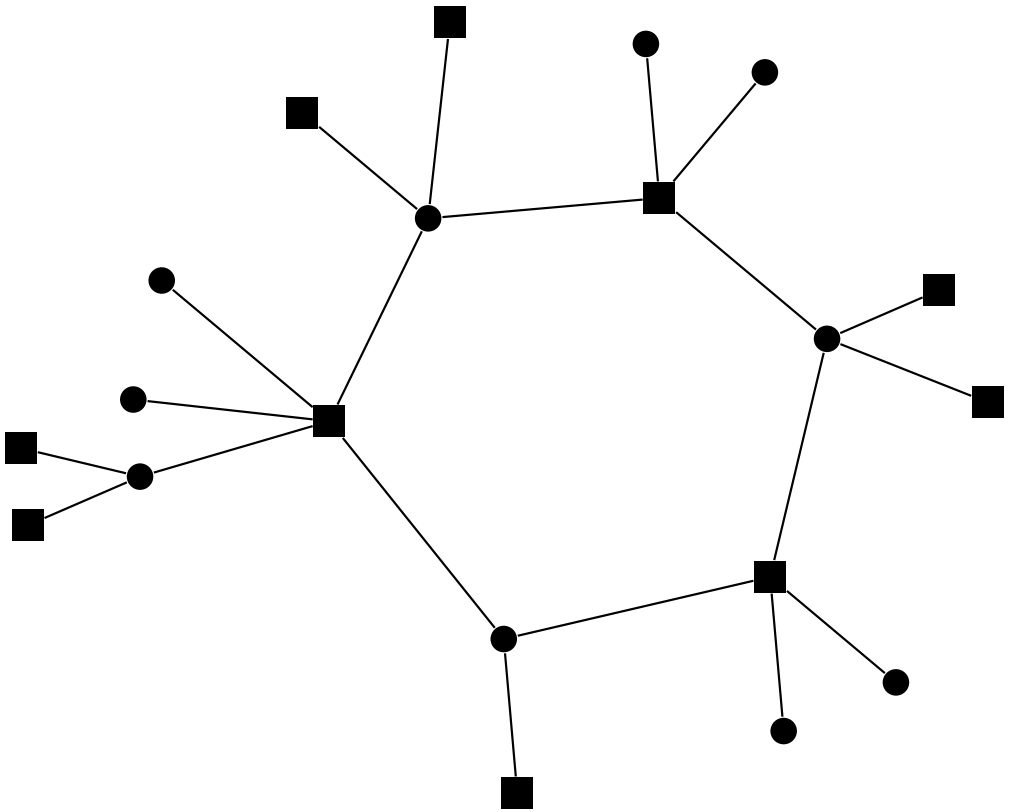
Given a vector (x_{ij}):

- $V = J \cup M$,
- $E_+ = \{(i,j) : 0 < x_{ij} < 1\}$,
- $G = (V, E_+)$.



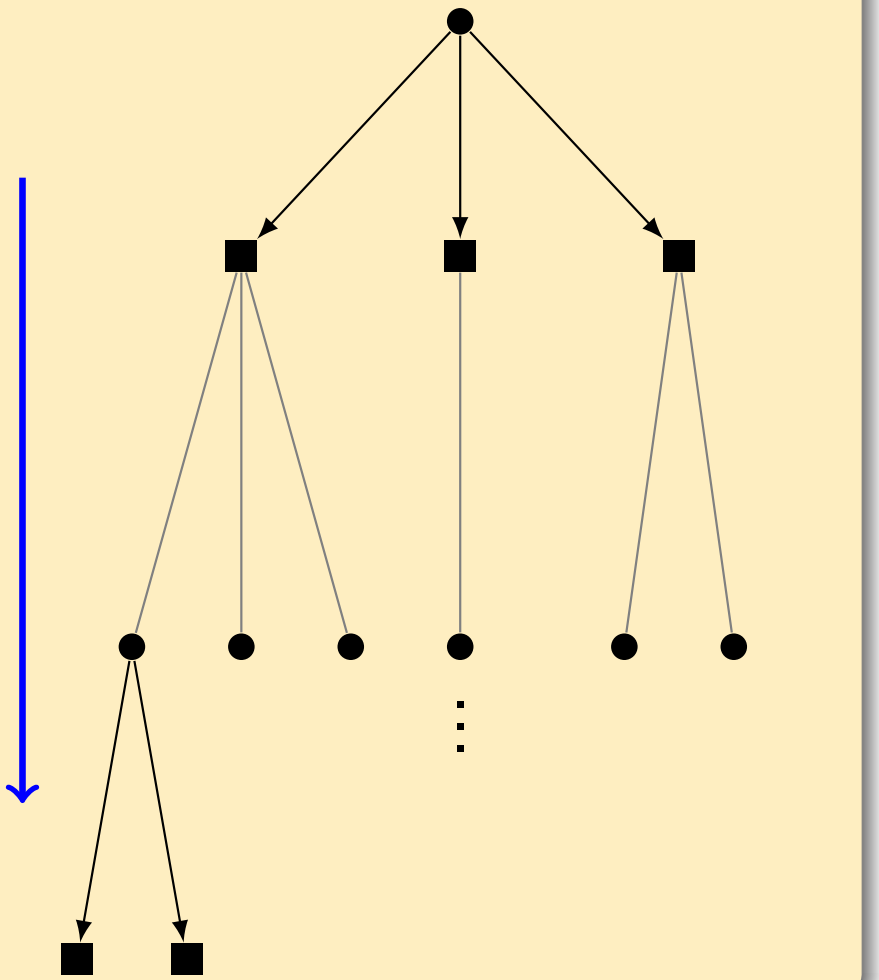
Property (Sparseness)

If x is an extreme solution to [LST]-LP then $G = (V, E_+)$ is a pseudo-forest.



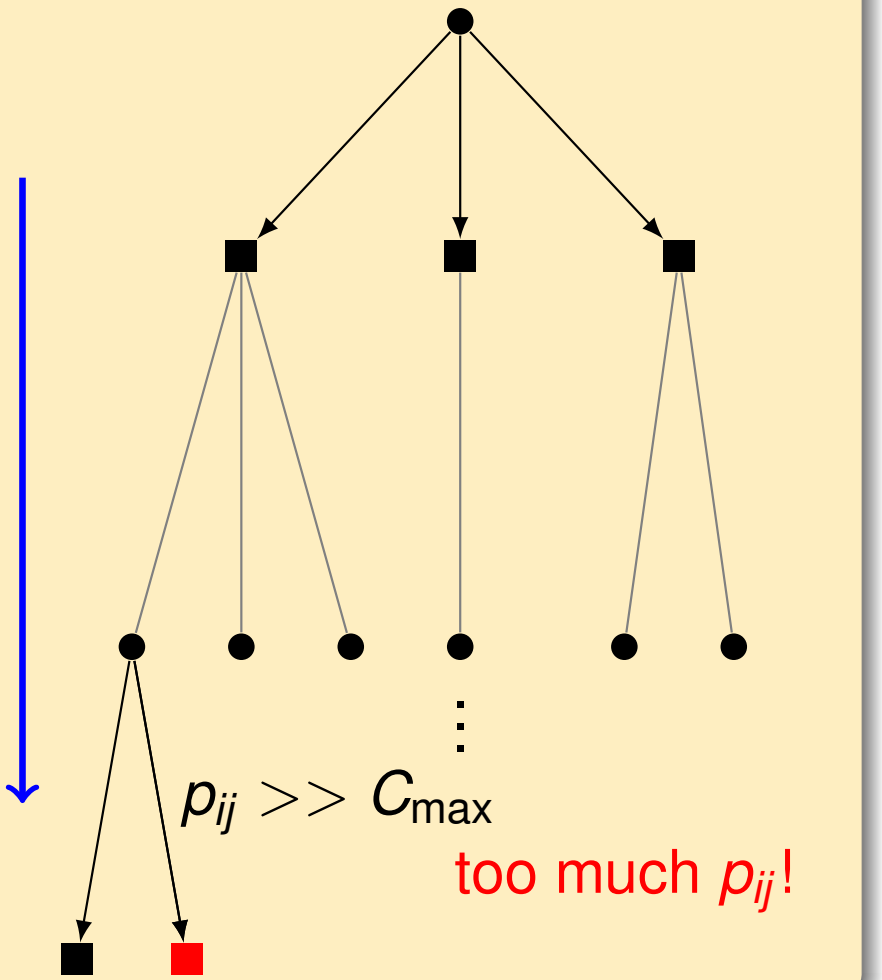
Difficulty

Don’t split (assign downwards)



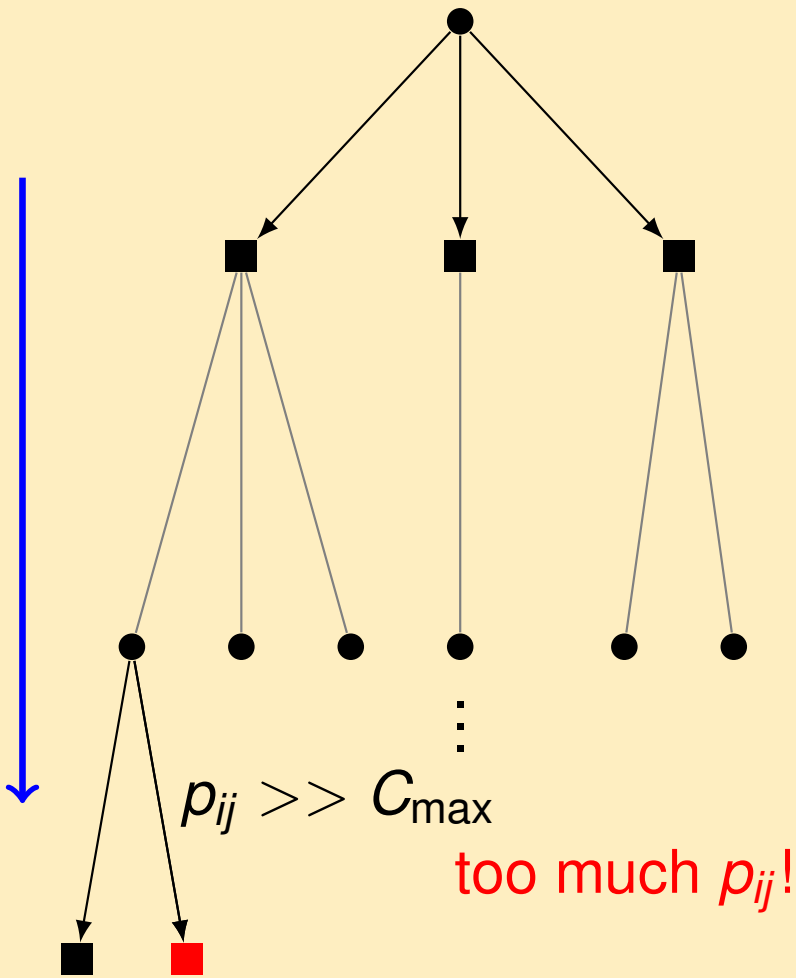
Difficulty

Don't split (assign downwards)

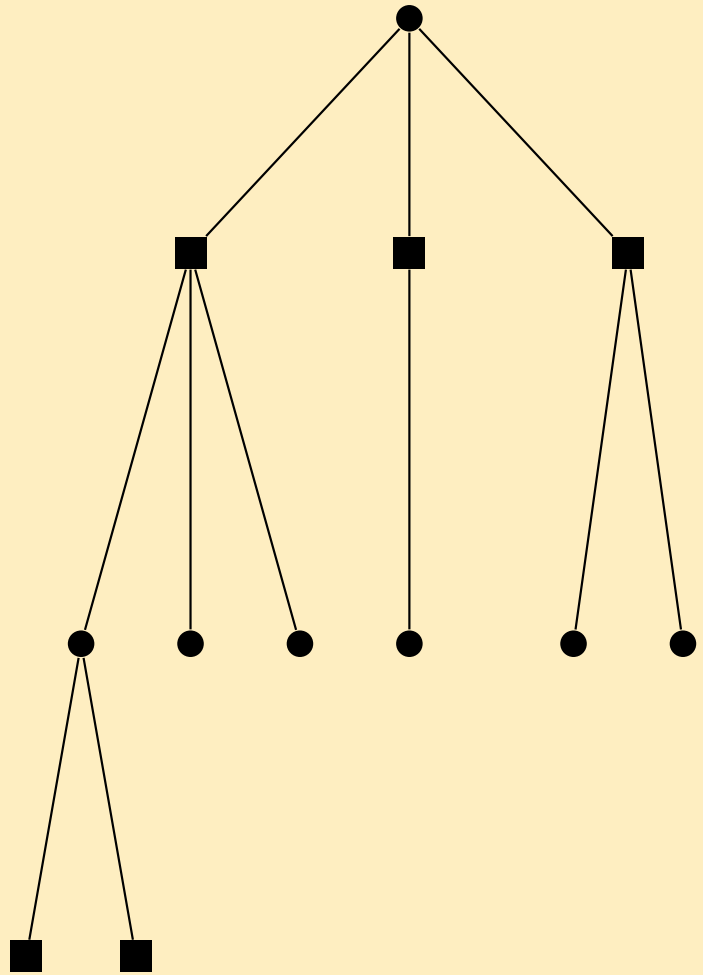


Difficulty

Don't split (assign downwards)

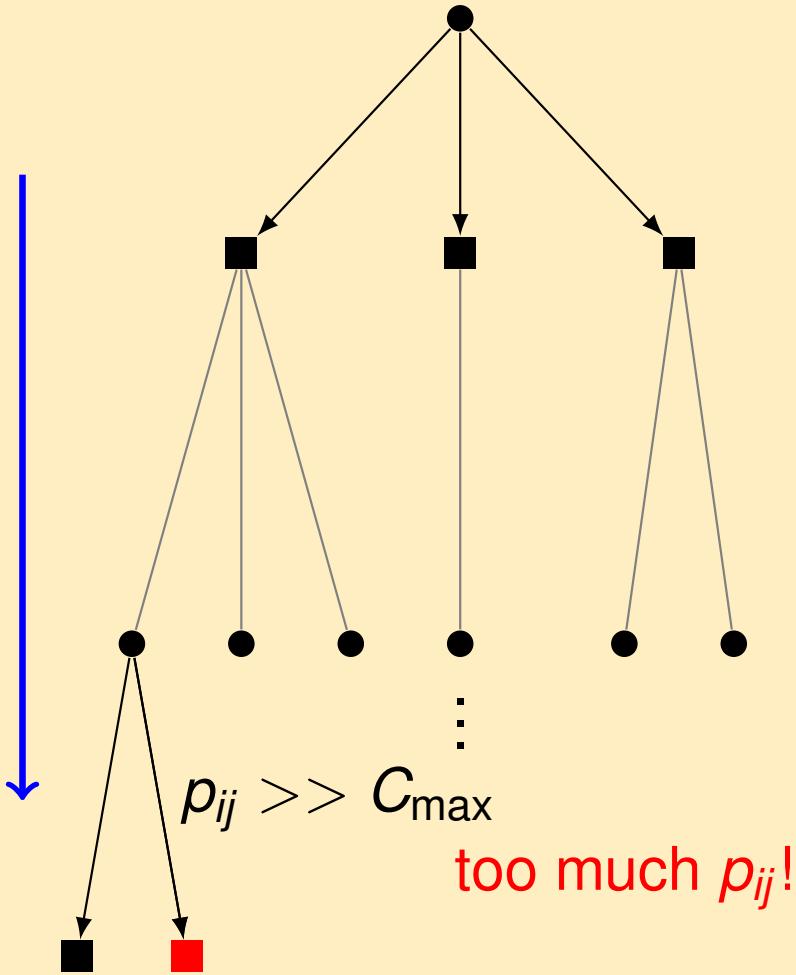


Split 100% (assign with x_{ij})

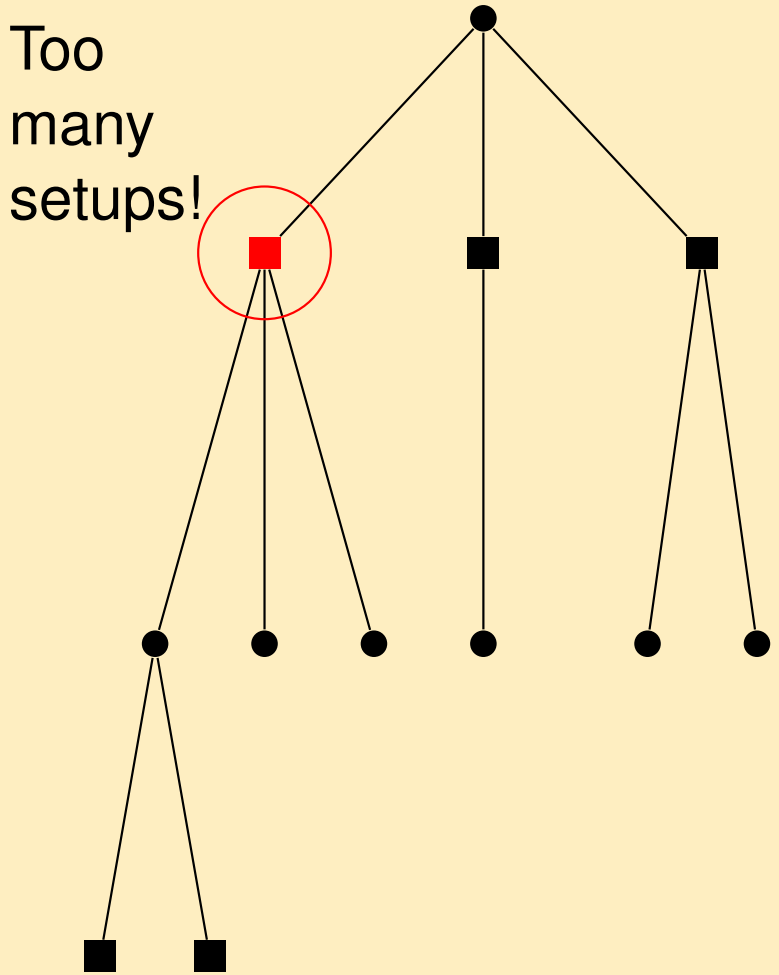


Difficulty

Don't split (assign downwards)



Split 100% (assign with x_{ij})



Rounding

Step 1: Heavy assignments

$$x_{ij} > \frac{1}{2} \quad \Rightarrow \quad \tilde{x}_{ij} = 1.$$

Rounding

Step 1: Heavy assignments

$$x_{ij} > \frac{1}{2} \quad \Rightarrow \quad \tilde{x}_{ij} = 1.$$

$$\sum_{j \in J} x_{ij}(p_{ij} + s_{ij}) \leq C_{\max}$$

Rounding

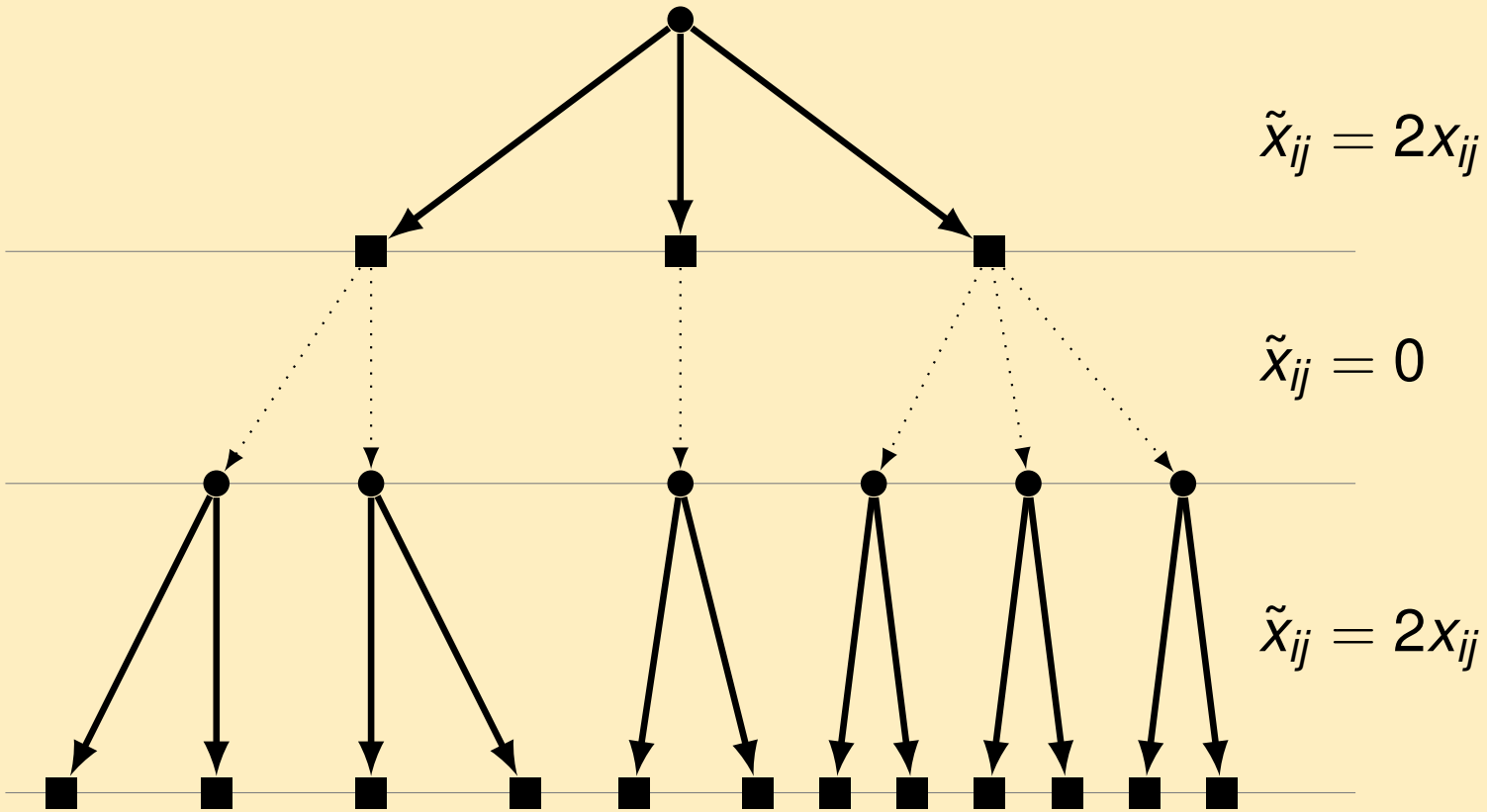
Step 2: Light assignments

$$E' := \left\{ (i,j) : 0 < x_{ij} < \frac{1}{2} \right\}.$$

Rounding

Step 2: Light assignments

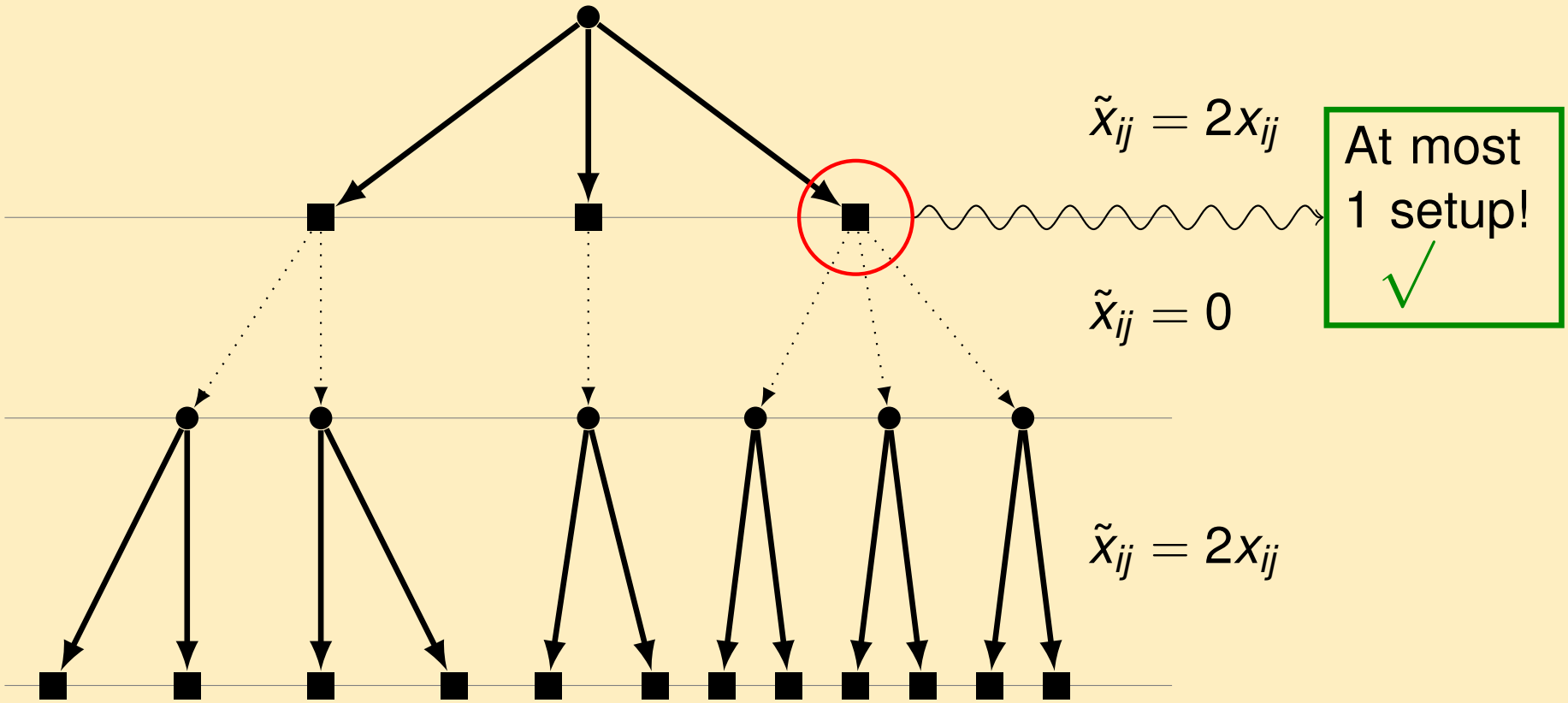
$$E' := \left\{ (i,j) : 0 < x_{ij} < \frac{1}{2} \right\}.$$



Rounding

Step 2: Light assignments

$$E' := \left\{ (i,j) : 0 < x_{ij} < \frac{1}{2} \right\}.$$



Result

Theorem

The rounding scheme yields a 3-approximation algorithm.

Proof.

By construction:

- All jobs are completely assigned.
- It holds

$$\begin{aligned}\text{load}(i) &\leq s_{ij^*} + \sum_{j \in J} \tilde{x}_{ij}(p_{ij} + s_{ij}) \\ &\leq s_{ij^*} + 2\sum_{j \in J} x_{ij}(p_{ij} + s_{ij}) \\ &\leq C_{\max} + 2C_{\max}.\end{aligned}$$

□

Best possible

Theorem

The [LST]-LP has a gap of 3.

Best possible

Theorem

The [LST]-LP has a gap of 3.

How to improve?

Improving the LP

Weakness

$$\sum_{j \in J} \left(x_{ij} p_{ij} + \underline{x_{ij} s_{ij}} \right) \leq C_{\max}.$$

Improving the LP

Weakness

$$\sum_{j \in J} \left(x_{ij} p_{ij} + \underline{x_{ij} s_{ij}} \right) \leq C_{\max}.$$

Add variables:

$$y_{ij} = \begin{cases} 1 & \text{if } j \text{ is assigned to } i, \\ 0 & \text{otherwise.} \end{cases}$$

Restricctions:

- $$x_{ij} \leq y_{ij}$$
- $$x_{ij} p_{ij} + s_{ij} y_{ij} \leq C_{\max} y_{ij} \quad \Rightarrow \quad x_{ij} \frac{p_{ij}}{C_{\max} - s_{ij}} \leq y_{ij}$$

New LP

Strong [LST]-LP

$$\sum_{i \in M} x_{ij} = 1 \qquad \text{for all } j,$$

$$\sum_{j \in J} \left(x_{ij} p_{ij} + y_{ij} s_{ij} \right) \leq C_{\max} \qquad \text{for all } i,$$

$$x_{ij} \leq y_{ij} \qquad \text{for all } i, j,$$

$$x_{ij} \frac{p_{ij}}{C_{\max} - s_{ij}} \leq y_{ij} \qquad \text{for all } i, j,$$

$$y_{ij} = 0 \qquad \text{if: } s_{ij} > C_{\max},$$

$$x_{ij} \geq 0.$$

New LP

Strong [LST]-LP

$$\sum_{i \in M} x_{ij} = 1 \qquad \text{for all } j,$$

$$\sum_{j \in J} \left(x_{ij} p_{ij} + x_{ij} s_{ij} \cdot \max \left\{ 1, \frac{p_{ij}}{C_{\max} - s_{ij}} \right\} \right) \leq C_{\max} \qquad \text{for all } i,$$

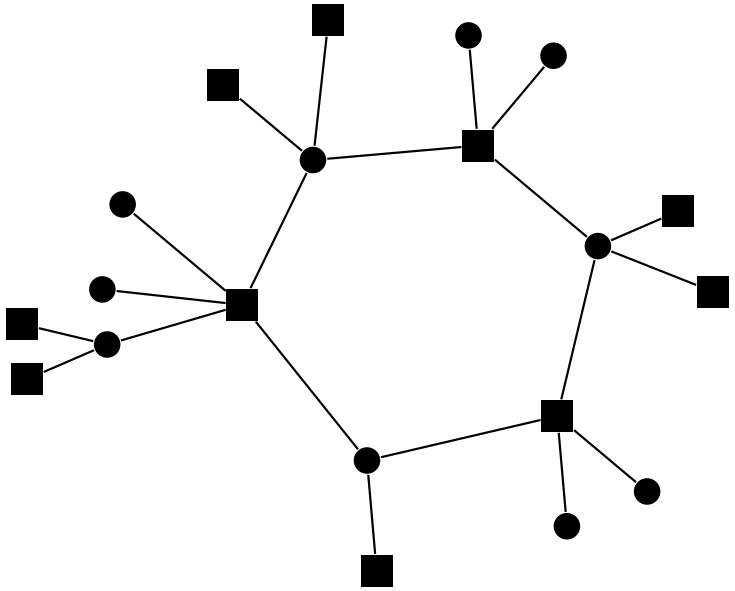
$$x_{ij} = 0 \qquad \text{if } s_{ij} > C_{\max},$$

$$x_{ij} \geq 0.$$

Rounding (again)

Property (Sparseness)

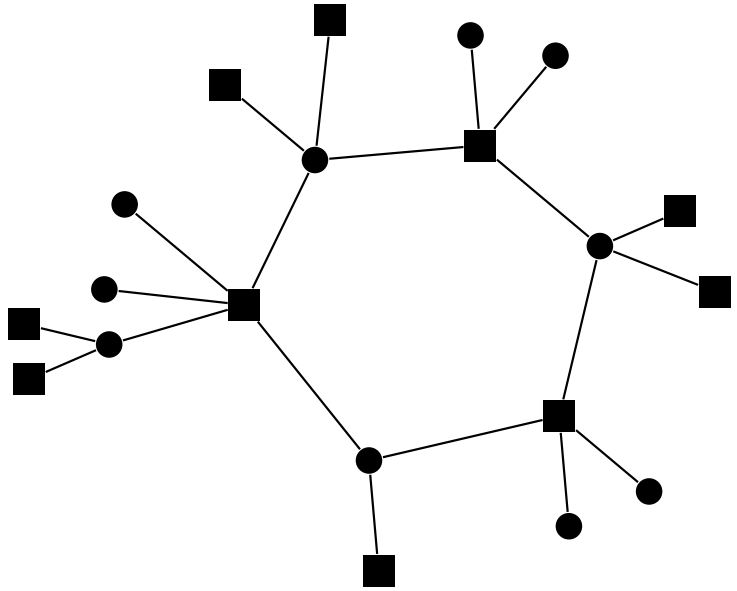
If x is an extreme solution to Strong-[LST] then $G = (V, E_+)$ is a pseudo-forest.



Rounding (again)

Property (Sparseness)

If x is an extreme solution to Strong-[LST] then $G = (V, E_+)$ is a pseudo-forest.



Step 1: Heavy assignments

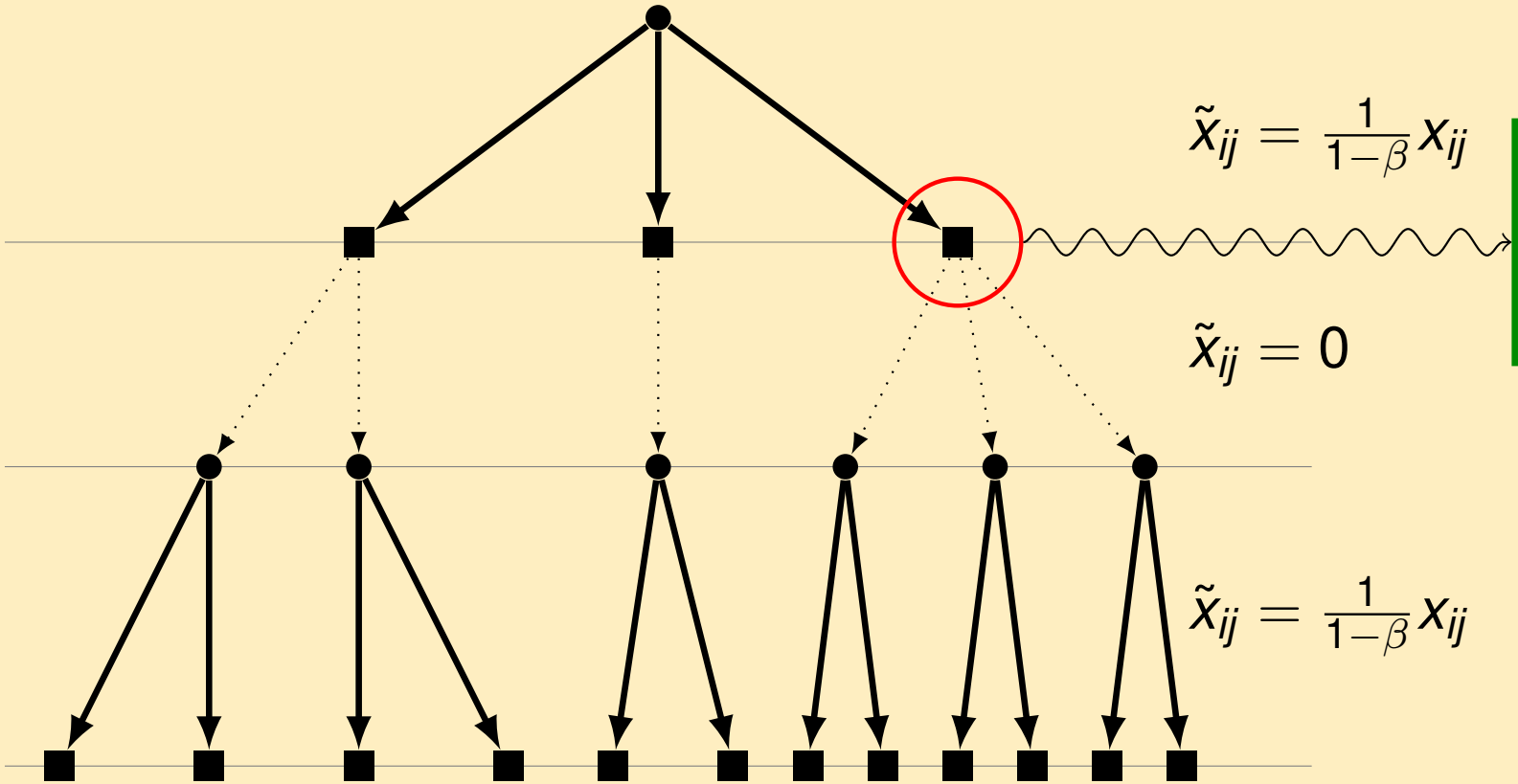
Given $1/2 < \beta < 1$:

$$x_{ij} > \beta \quad \Rightarrow \quad \tilde{x}_{ij} = 1.$$

Rounding

Step 2: Light assignments

$$E' := \{(i,j) : 0 < x_{ij} < \beta\}.$$



At most 1 setup! ✓

Result

Theorem

The rounding scheme yields an approximation ratio of $1 + \phi \approx 2.618$.

Proof idea.

$$\text{load}(i) \leq \tilde{x}_{ij^*} p_{ij^*} + s_{ij^*} + \sum_{x_{ij} > \beta} \tilde{x}_{ij}(p_{ij} + s_{ij}).$$

Recall that

$$\sum_{j \in J} \left(x_{ij} p_{ij} + x_{ij} s_{ij} \cdot \frac{p_{ij}}{C_{\max} - s_{ij}} \right) \leq C_{\max}$$

$$\vdots \quad \text{balancing}$$

$$\text{load}(i) \leq \max \left\{ \frac{1}{1-\beta}, 1 + \frac{1}{\beta} \right\} \cdot C_{\max} \ldots$$

Result

Theorem

The rounding scheme yields an approximation ratio of $1 + \phi \approx 2.618$.

Proof idea.

$$\text{load}(i) \leq \tilde{x}_{ij^*} p_{ij^*} + s_{ij^*} + \sum_{x_{ij} > \beta} \tilde{x}_{ij}(p_{ij} + s_{ij}).$$

Recall that

$$\sum_{j \in J} \left(x_{ij} p_{ij} + x_{ij} p_{ij} \cdot \frac{1}{C_{\max}/s_{ij} - 1} \right) \leq C_{\max}$$

$$\vdots \quad \text{balancing}$$

$$\text{load}(i) \leq \max \left\{ \frac{1}{1-\beta}, 1 + \frac{1}{\beta} \right\} \cdot C_{\max} \ldots$$

More results

Theorem

Strong-[LST] has a gap of $1 + \phi$.

More results

Theorem

Strong-[LST] has a gap of $1 + \phi$.

Theorem

The problem is NP-hard to approximate within $\frac{e}{e-1} \approx 1.581$.

More results

Theorem

Strong-[LST] has a gap of $1 + \phi$.

Theorem

The problem is NP-hard to approximate within $\frac{e}{e-1} \approx 1.581$.

Theorem

For restricted assignment ($s_{ij} \in \{s_j, \infty\}, p_{ij} = p_j$) *there exists a 2-approximation algorithm.*

More results

Theorem

Strong-[LST] has a gap of $1 + \phi$.

Theorem

The problem is NP-hard to approximate within $\frac{e}{e-1} \approx 1.581$.

Theorem

For restricted assignment ($s_{ij} \in \{s_j, \infty\}, p_{ij} = p_j$) *there exists a 2-approximation algorithm.*

Can we improve further?!

Preliminaries

Given $C_{\max}$,

- Set $\mathcal{C}_j$: all possible *configurations* for job j.
- For $c \in \mathcal{C}_j$,
  - $t_{c,i}$: time used by job j on machine i if j uses c,
  - $\lambda_c \in \{0, 1\}$: indicates if j uses conf. c.

[CLP]:

$$\sum_{c \in \mathcal{C}_j} \lambda_c = 1 \qquad \text{for every } j \in J,$$

$$\sum_{j \in J} \sum_{c \in \mathcal{C}_j} \lambda_c \cdot t_{c,i} \leq C_{\max} \qquad \text{for every } i \in M,$$

$$\lambda_c \geq 0.$$