Approximation Algorithms

Lecture 7: Scheduling Jobs on Parallel Machines

Part I: ILP & Parametric Pruning

Joachim Spoerhase

Winter 2020/21

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,

$$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,
a set $\mathcal{M}$ of **machines**, and

$$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$$

$$\mathcal{M} = \{M_1, M_2, M_3\}$$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,
a set $\mathcal{M}$ of **machines**, and
for each $M_i \in \mathcal{M}$ and $J_j \in \mathcal{J}$ the **processing time** $p_{ij} \in \mathbb{N}^+$ of J_j on M_i.

$$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$$

$$\mathcal{M} = \{M_1, M_2, M_3\}$$

$$(p_{ij})_{M_i \in M, J_j \in J}$$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,
a set $\mathcal{M}$ of **machines**, and
for each $M_i \in \mathcal{M}$ and $J_j \in \mathcal{J}$ the **processing time** $p_{ij} \in \mathbb{N}^+$ of J_j on M_i.

Task: A **schedule** $\sigma \colon \mathcal{J} \to \mathcal{M}$ of the jobs on the machines which minimizes the total time to completion (**makespan**), i.e., minimizes the maximum time a machine is in use.

$$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$$

$$\mathcal{M} = \{M_1, M_2, M_3\}$$

$$(p_{ij})_{M_i \in M, J_j \in J}$$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,
a set $\mathcal{M}$ of **machines**, and
for each $M_i \in \mathcal{M}$ and $J_j \in \mathcal{J}$ the **processing time** $p_{ij} \in \mathbb{N}^+$ of J_j on M_i.

Task: A **schedule** $\sigma \colon \mathcal{J} \to \mathcal{M}$ of the jobs on the machines which minimizes the total time to completion (**makespan**), i.e., minimizes the maximum time a machine is in use.

M_1

M_2

M_3

$$\mathcal{J} = \{J_1, J_2, \dots, J_8\}$$

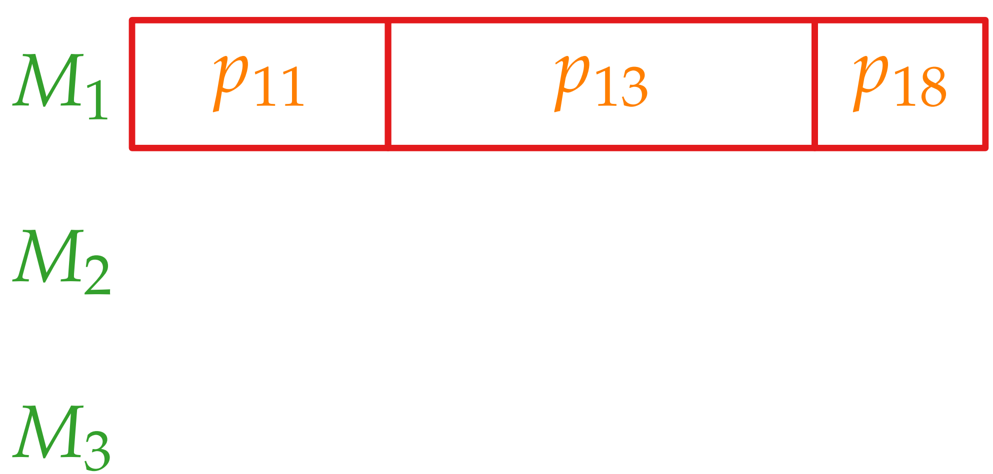
$$\mathcal{M} = \{M_1, M_2, M_3\}$$

$$(p_{ij})_{M_i \in M, J_j \in J}$$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,
a set $\mathcal{M}$ of **machines**, and
for each $M_i \in \mathcal{M}$ and $J_j \in \mathcal{J}$ the **processing time** $p_{ij} \in \mathbb{N}^+$ of J_j on M_i.

Task: A **schedule** $\sigma\colon \mathcal{J} \to \mathcal{M}$ of the jobs on the machines which minimizes the total time to completion (**makespan**), i.e., minimizes the maximum time a machine is in use.



$$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$$

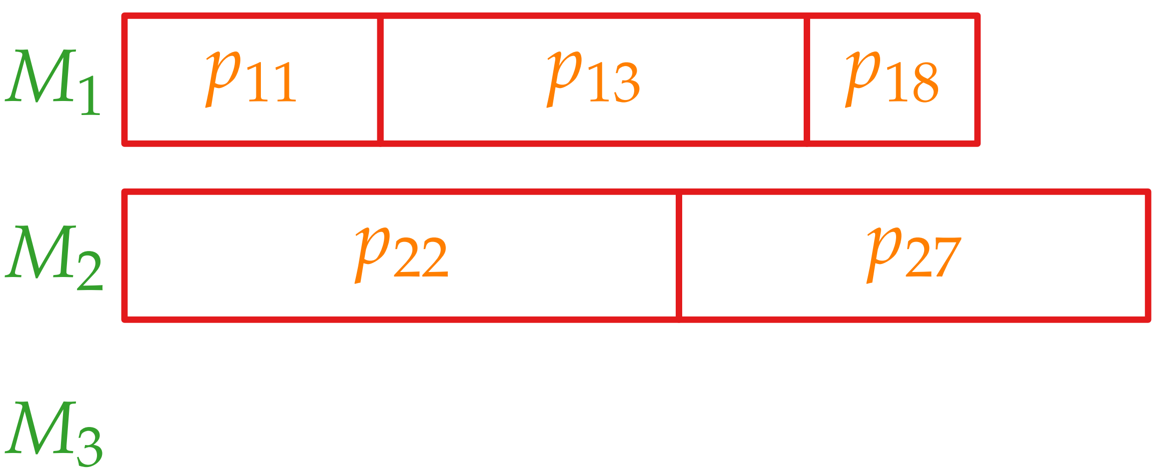
$$\mathcal{M} = \{M_1, M_2, M_3\}$$

$$(p_{ij})_{M_i \in M, J_j \in J}$$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,
a set $\mathcal{M}$ of **machines**, and
for each $M_i \in \mathcal{M}$ and $J_j \in \mathcal{J}$ the **processing time** $p_{ij} \in \mathbb{N}^+$ of J_j on M_i.

Task: A **schedule** $\sigma\colon \mathcal{J} \to \mathcal{M}$ of the jobs on the machines which minimizes the total time to completion (**makespan**), i.e., minimizes the maximum time a machine is in use.



$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$

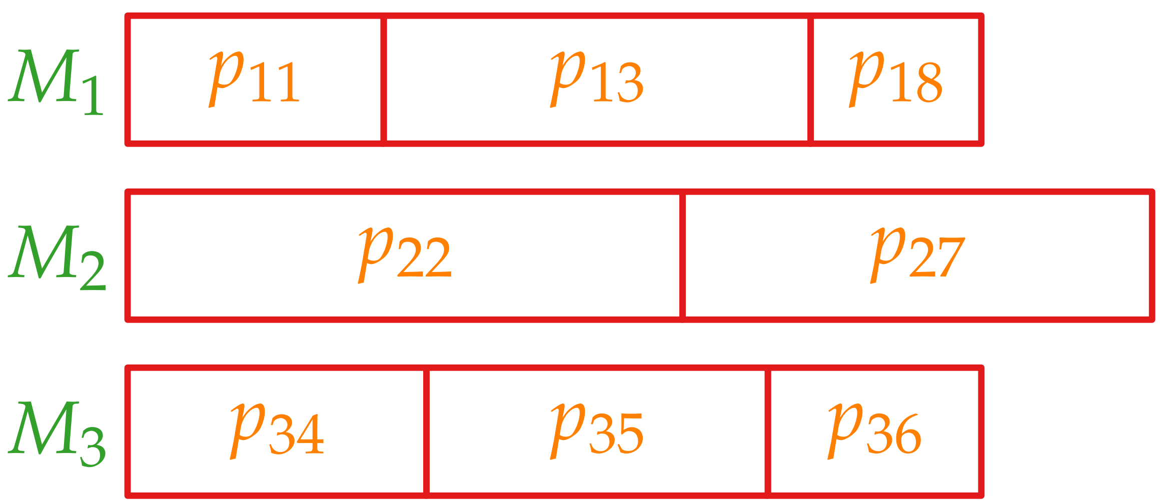
$\mathcal{M} = \{M_1, M_2, M_3\}$

$(p_{ij})_{M_i \in M, J_j \in J}$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**,
a set $\mathcal{M}$ of **machines**, and
for each $M_i \in \mathcal{M}$ and $J_j \in \mathcal{J}$ the **processing time** $p_{ij} \in \mathbb{N}^+$ of J_j on M_i.

Task: A **schedule** $\sigma \colon \mathcal{J} \to \mathcal{M}$ of the jobs on the machines which minimizes the total time to completion (**makespan**), i.e., minimizes the maximum time a machine is in use.



$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$

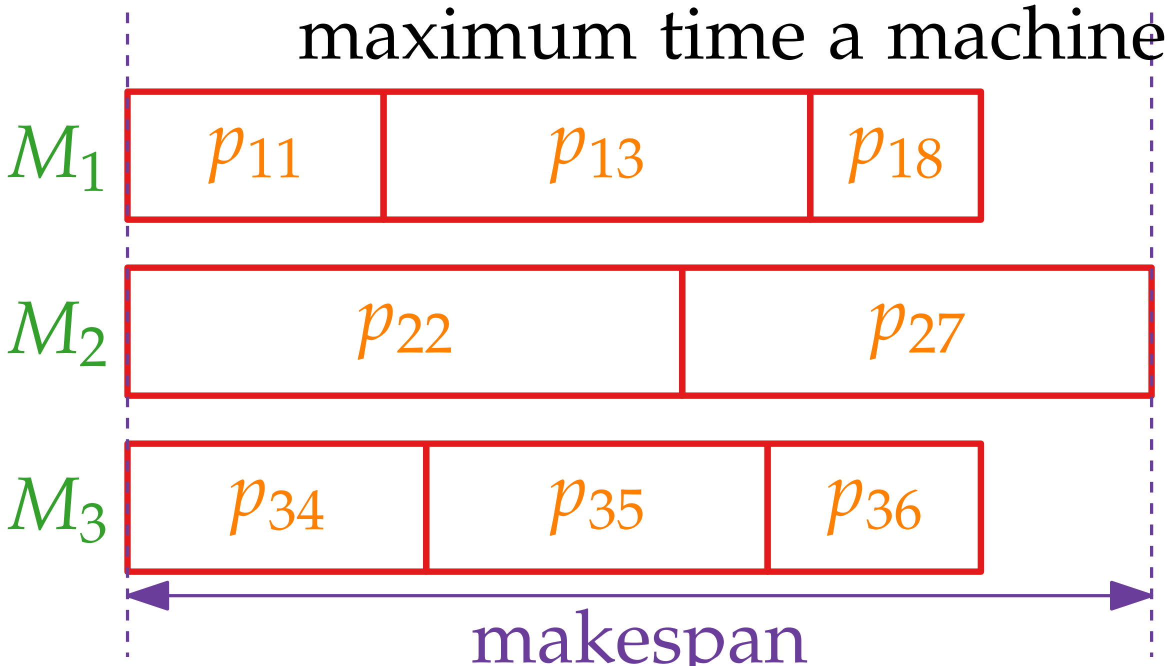
$\mathcal{M} = \{M_1, M_2, M_3\}$

$(p_{ij})_{M_i \in M, J_j \in J}$

Scheduling on Parallel Machines

Given: A set $\mathcal{J}$ of **jobs**, a set $\mathcal{M}$ of **machines**, and for each $M_i \in \mathcal{M}$ and $J_j \in \mathcal{J}$ the **processing time** $p_{ij} \in \mathbb{N}^+$ of J_j on M_i.

Task: A **schedule** $\sigma \colon \mathcal{J} \to \mathcal{M}$ of the jobs on the machines which minimizes the total time to completion (**makespan**), i.e., minimizes the maximum time a machine is in use.



$$\mathcal{J} = \{J_1, J_2, \ldots, J_8\}$$

$$\mathcal{M} = \{M_1, M_2, M_3\}$$

$$(p_{ij})_{M_i \in M, J_j \in J}$$

Formulation as ILP

minimize t

subject to

Formulation as ILP

minimize t

subject to

$$x_{ij} \in \{0,1\}, \qquad M_i \in \mathcal{M}, J_j \in \mathcal{J}$$

Formulation as ILP

minimize t

subject to $\qquad J_j \in \mathcal{J}$

$\qquad M_i \in \mathcal{M}$

$x_{ij} \in \{0,1\}, \qquad M_i \in \mathcal{M}, J_j \in \mathcal{J}$

Formulation as ILP

$$
\begin{aligned}
\textbf{minimize} \quad & t \\
\textbf{subject to} \quad & \sum_{M_i \in \mathcal{M}} x_{ij} = 1, && J_j \in \mathcal{J} \\
& && M_i \in \mathcal{M} \\
& x_{ij} \in \{0,1\}, && M_i \in \mathcal{M}, J_j \in \mathcal{J}
\end{aligned}
$$

Formulation as ILP

$$
\begin{array}{lll}
\textbf{minimize} & t & \\
\textbf{subject to} & \displaystyle\sum_{M_i \in \mathcal{M}} x_{ij} = 1, & J_j \in \mathcal{J} \\
& \displaystyle\sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, & M_i \in \mathcal{M} \\
& x_{ij} \in \{0,1\}, & M_i \in \mathcal{M}, J_j \in \mathcal{J}
\end{array}
$$

Formulation as ILP

$$
\begin{aligned}
\textbf{minimize} \quad & t \\
\textbf{subject to} \quad & \sum_{M_i \in \mathcal{M}} x_{ij} = 1, && J_j \in \mathcal{J} \\
& \sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, && M_i \in \mathcal{M} \\
& x_{ij} \in \{0,1\}, && M_i \in \mathcal{M}, J_j \in \mathcal{J}
\end{aligned}
$$

Task: Prove that the integrality gap is unbounded!

Formulation as ILP

$$
\begin{aligned}
\textbf{minimize} \quad & t \\
\textbf{subject to} \quad & \sum_{M_i \in \mathcal{M}} x_{ij} = 1, && J_j \in \mathcal{J} \\
& \sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, && M_i \in \mathcal{M} \\
& x_{ij} \in \{0, 1\}, && M_i \in \mathcal{M}, J_j \in \mathcal{J}
\end{aligned}
$$

Task: Prove that the integrality gap is unbounded!

Solution: m machines and one job with processing time m

Formulation as ILP

$$\begin{array}{lll} \textbf{minimize} & t & \\ \textbf{subject to} & \sum_{M_i \in \mathcal{M}} x_{ij} = 1, & J_j \in \mathcal{J} \\ & \sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, & M_i \in \mathcal{M} \\ & x_{ij} \in \{0,1\}, & M_i \in \mathcal{M}, J_j \in \mathcal{J} \end{array}$$

Task: Prove that the integrality gap is unbounded!

Solution: m machines and one job with processing time m

$\Rightarrow$ OPT $= m$ and $\text{OPT}_{\text{frac}} = 1$.

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) \colon M_i \in \mathcal{M},\, J_j \in \mathcal{J},\, p_{ij} \leq T \,\}$.

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) : M_i \in \mathcal{M},\, J_j \in \mathcal{J},\, p_{ij} \leq T \,\}$.

Define the "pruned" relaxation LP(T):

$$\begin{aligned}
\textbf{minimize} \quad & t \\
\textbf{subject to} \quad & \sum_{M_i \in \mathcal{M}} x_{ij} = 1, && J_j \in \mathcal{J} \\
& \sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, && M_i \in \mathcal{M} \\
& x_{ij} \in \{0, 1\}, && M_i \in \mathcal{M}, J_j \in \mathcal{J}
\end{aligned}$$

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{ (i,j) : M_i \in \mathcal{M}, J_j \in \mathcal{J}, p_{ij} \leq T \}$.

Define the "pruned" relaxation LP(T):

$$\begin{aligned} \textbf{minimize} \quad & t \\ \textbf{subject to} \quad & \sum_{M_i \in \mathcal{M}} x_{ij} = 1, \quad J_j \in \mathcal{J} \\ & \sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, \quad M_i \in \mathcal{M} \\ & x_{ij} \, \cancel{\in \{0,1\}}, \geq 0 \quad M_i \in \mathcal{M}, J_j \in \mathcal{J} \end{aligned}$$

Paremetric Pruning

Strengthen the ILP $\to$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) : M_i \in \mathcal{M},\ J_j \in \mathcal{J},\ p_{ij} \leq T \,\}$.

Define the "pruned" relaxation LP(T):

minimize t

subject to

$$\sum_{M_i \in \mathcal{M}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, \qquad M_i \in \mathcal{M}$$

x_{ij} ~~$\in \{0,1\}$~~, ≥ 0 ~~$M_i \in \mathcal{M}, J_j \in \mathcal{J}$~~ $(i,j) \in S_T$

Paremetric Pruning

Strengthen the ILP $\to$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) \colon M_i \in \mathcal{M},\ J_j \in \mathcal{J},\ p_{ij} \leq T \,\}$.

Define the "pruned" relaxation LP(T):

$$\begin{array}{lll} \textbf{minimize} & t & \\ \textbf{subject to} & \displaystyle\sum_{(i,j) \in S_T\ \cancel{M_i \in \mathcal{M}}} x_{ij} = 1, & J_j \in \mathcal{J} \\ & \displaystyle\sum_{J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t, & M_i \in \mathcal{M} \\ & x_{ij}\ \cancel{\in \{0,1\}},\ \geq 0 \quad \cancel{M_i \in \mathcal{M}, J_j \in \mathcal{J}} & (i,j) \in S_T \end{array}$$

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{ (i,j) : M_i \in \mathcal{M}, J_j \in \mathcal{J}, p_{ij} \leq T \}$.

Define the "pruned" relaxation LP(T):

$$
\begin{aligned}
\textbf{minimize} \quad & t \\
\textbf{subject to} \quad & \sum_{(i,j) \in S_T \; \cancel{M_i \in \mathcal{M}}} x_{ij} = 1, \quad J_j \in \mathcal{J} \\
& \sum_{(i,j) \in S_T \; \cancel{J_j \in \mathcal{J}}} x_{ij} p_{ij} \leq t, \quad M_i \in \mathcal{M} \\
& x_{ij} \cancel{\in \{0,1\}}, \geq 0 \quad \cancel{M_i \in \mathcal{M}, J_j \in \mathcal{J}} \quad (i,j) \in S_T
\end{aligned}
$$

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) \colon M_i \in \mathcal{M},\ J_j \in \mathcal{J},\ p_{ij} \leq T \,\}$.

Define the "pruned" relaxation LP(T):

minimize $\quad t$

subject to

$$\sum_{(i,j) \in S_T \ \cancel{M_i \in \mathcal{M}}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_T \ \cancel{J_j \in \mathcal{J}}} x_{ij} p_{ij} \leq \cancel{t,}\ T \quad M_i \in \mathcal{M}$$

$$x_{ij} \ \cancel{\in \{0,1\},} \geq 0 \quad \cancel{M_i \in \mathcal{M}, J_j \in \mathcal{J}} \quad (i,j) \in S_T$$

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) \colon M_i \in \mathcal{M},\ J_j \in \mathcal{J},\ p_{ij} \leq T \,\}$.

Define the "pruned" relaxation LP(T):

~~**minimize** t~~

subject to

$$\sum_{(i,j) \in S_T\ \cancel{M_i \in \mathcal{M}}} x_{ij} = 1, \quad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_T\ \cancel{J_j \in \mathcal{J}}} x_{ij} p_{ij} \leq \cancel{t,}\ T \quad M_i \in \mathcal{M}$$

$$x_{ij}\ \cancel{\in \{0,1\}},\ \geq 0 \quad \cancel{M_i \in \mathcal{M}, J_j \in \mathcal{J}}\quad (i,j) \in S_T$$

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) \colon M_i \in \mathcal{M},\, J_j \in \mathcal{J},\, p_{ij} \le T \,\}$.

Define the "pruned" relaxation LP(T):

$$\begin{aligned} \sum_{(i,j) \in S_T} x_{ij} &= 1, && J_j \in \mathcal{J} \\ \sum_{(i,j) \in S_T} x_{ij} p_{ij} &\le T, && M_i \in \mathcal{M} \\ x_{ij} &\ge 0, && (i,j) \in S_T \end{aligned}$$

Paremetric Pruning

Strengthen the ILP $\rightarrow$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{\, (i,j) : M_i \in \mathcal{M},\, J_j \in \mathcal{J},\, p_{ij} \leq T \,\}$.

Define the "pruned" relaxation LP(T):

$$
\begin{aligned}
\sum_{(i,j)\in S_T} x_{ij} &= 1, && J_j \in \mathcal{J} \\
\sum_{(i,j)\in S_T} x_{ij} p_{ij} &\leq T, && M_i \in \mathcal{M} \\
x_{ij} &\geq 0, && (i,j) \in S_T
\end{aligned}
$$

LP(T) has no objective function; we just need to determine if a feasible solution exists.

Paremetric Pruning

Strengthen the ILP $\to$ implicit (non-linear) constraint:

If $p_{ij} > t$, then set $x_{ij} = 0$.

Introduce new parameter $T \in \mathbb{N}$ to estimate a lower bound on OPT.

Define $S_T := \{ (i,j) \colon M_i \in \mathcal{M}, J_j \in \mathcal{J}, p_{ij} \leq T \}$.

Define the "pruned" relaxation LP(T):

$$\begin{aligned} \sum_{(i,j) \in S_T} x_{ij} &= 1, && J_j \in \mathcal{J} \\ \sum_{(i,j) \in S_T} x_{ij} p_{ij} &\leq T, && M_i \in \mathcal{M} \\ x_{ij} &\geq 0, && (i,j) \in S_T \end{aligned}$$

LP(T) has no objective function; we just need to determine if a feasible solution exists.

But why does this LP give a good integrality gap?

Approximation Algorithms

Lecture 7:
Scheduling Jobs on Parallel Machines

Part II:
Properties of Extreme Point Solutions

Joachim Spoerhase

Winter 2020/21

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that LP(T) has a solution.

LP(T)

$$\begin{aligned} \sum_{(i,j)\in S_T} x_{ij} &= 1, & J_j &\in \mathcal{J} \\ \sum_{(i,j)\in S_T} x_{ij} p_{ij} &\le T, & M_i &\in \mathcal{M} \\ x_{ij} &\ge 0, & (i,j) &\in S_T \end{aligned}$$

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that LP(T) has a solution. Let T^* be this value of T.

LP(T)

$$\begin{aligned}
\sum_{(i,j)\in S_T} x_{ij} &= 1, && J_j \in \mathcal{J} \\
\sum_{(i,j)\in S_T} x_{ij} p_{ij} &\le T, && M_i \in \mathcal{M} \\
x_{ij} &\ge 0, && (i,j) \in S_T
\end{aligned}$$

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that LP(T) has a solution. Let T^* be this value of T.

What are the bounds for our search?

LP(T)

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that LP(T) has a solution. Let T^* be this value of T.

What are the bounds for our search?

Observe: $T^* \leq \text{OPT}$

LP(T)

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that LP(T) has a solution. Let T^* be this value of T.

What are the bounds for our search?

Observe: $T^* \leq \mathrm{OPT}$

Idea: Round an extreme-point solution of LP(T^*) to a schedule whose makespan is $\leq 2T^*$

LP(T)

$$\sum_{(i,j) \in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that LP(T) has a solution. Let T^* be this value of T.

What are the bounds for our search?

Observe: $T^* \leq \mathrm{OPT}$

Idea: Round an extreme-point solution of LP(T^*) to a schedule whose makespan is $\leq 2T^*$

Lemma 1.
Each extreme point solution for LP(T) has $\leq |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

LP(T)

$$\sum_{(i,j) \in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that LP(T) has a solution. Let T^* be this value of T.

What are the bounds for our search?

Observe: $T^* \leq \text{OPT}$

Idea: Round an extreme-point solution of LP(T^*) to a schedule whose makespan is $\leq 2T^*$

LP(T)

$$\sum_{(i,j) \in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\leq |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Lemma 1

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \le T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \ge 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\le |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Lemma 1

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij}p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\leq |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Proof. $L(T)$: $|S_T|$ variables

Lemma 1

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \le T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \ge 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\le |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Proof. $L(T)$: $|S_T|$ variables
extreme point sol.: $|S_T|$ inequalities tight

Lemma 1

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\leq |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Proof. $L(T)$: $|S_T|$ variables
extreme point sol.: $|S_T|$ inequalities tight
max. $|\mathcal{J}|$

Lemma 1

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \le T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \ge 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\le |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Proof. $L(T)$: $|S_T|$ variables
extreme point sol.: $|S_T|$ inequalities tight

max. $|\mathcal{J}|$
max. $|\mathcal{M}|$

Lemma 1

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\leq |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Proof. $L(T)$: $|S_T|$ variables
extreme point sol.: $|S_T|$ inequalities tight
max. $|\mathcal{J}|$
max. $|\mathcal{M}|$
$\Rightarrow$ min. $|S_T| - |\mathcal{J}| - |\mathcal{M}|$

Lemma 1

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \le T, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \ge 0, \qquad (i,j) \in S_T$$

Lemma 1.
Each extreme point solution for LP(T) has $\le |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

Proof. $L(T)$: $|S_T|$ variables
extreme point sol.: $|S_T|$ inequalities tight

max. $|\mathcal{J}|$
max. $|\mathcal{M}|$
$\Rightarrow$ min. $|S_T| - |\mathcal{J}| - |\mathcal{M}|$
$\Rightarrow$ max. $|\mathcal{M}| + |\mathcal{J}|$ not night $\square$

Lemma 2

$$\sum_{(i,j) \in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j) \in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Lemma 2

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij}p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.

Lemma 2

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.
Assume α jobs integral und β jobs fractional in x.

Lemma 2

$$\sum_{(i,j) \in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j) \in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.
Assume α jobs integral und β jobs fractional in x.
$\Rightarrow \alpha + \beta = |\mathcal{J}|$

Lemma 2

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij}p_{ij} \le T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \ge 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\ge |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.
Assume α jobs integral und β jobs fractional in x.
$\Rightarrow \alpha + \beta = |\mathcal{J}|$
Fractional jobs: ≥ 2 machines

Lemma 2

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.
Assume α jobs integral und β jobs fractional in x.
$\Rightarrow \alpha + \beta = |\mathcal{J}|$
Fractional jobs: ≥ 2 machines
$\Rightarrow \geq 2$ variables > 0

Lemma 2

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \quad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \le T, \quad M_i \in \mathcal{M}$$
$$x_{ij} \ge 0, \quad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\ge |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.
Assume α jobs integral und β jobs fractional in x.
$\Rightarrow \alpha + \beta = |\mathcal{J}|$
Fractional jobs: ≥ 2 machines
$\Rightarrow \ge 2$ variables > 0
$\Rightarrow \alpha + 2\beta \le |\mathcal{J}| + |\mathcal{M}|$ (Lemma 1)

Lemma 2

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.
Assume α jobs integral und β jobs fractional in x.
$\Rightarrow \alpha + \beta = |\mathcal{J}|$
Fractional jobs: ≥ 2 machines
$\Rightarrow \geq 2$ variables > 0
$\Rightarrow \alpha + 2\beta \leq |\mathcal{J}| + |\mathcal{M}|$ (Lemma 1)
$\Rightarrow \beta \leq |\mathcal{M}|$

Lemma 2

$$\sum_{(i,j)\in S_T} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$
$$\sum_{(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad M_i \in \mathcal{M}$$
$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Lemma 2.
Any extreme point solution for LP(T) must set $\geq |\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be extreme point solution for $L(T)$.
Assume α jobs integral und β jobs fractional in x.
$\Rightarrow \alpha + \beta = |\mathcal{J}|$
Fractional jobs: ≥ 2 machines
$\Rightarrow \geq 2$ variables > 0
$\Rightarrow \alpha + 2\beta \leq |\mathcal{J}| + |\mathcal{M}|$ (Lemma 1)
$\Rightarrow \beta \leq |\mathcal{M}| \quad \Rightarrow \alpha \geq |\mathcal{J}| - |\mathcal{M}|$ □

Approximation Algorithms

Lecture 7:
Scheduling Jobs on Parallel Machines

Part III:
An Algorithm

Joachim Spoerhase

Winter 2020/21

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.

$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.
$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$

Let $F \subseteq \mathcal{J}$ be the set of fractionally assigned jobs.

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.

$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$

Let $F \subseteq \mathcal{J}$ be the set of fractionally assigned jobs.
Let $H := G[\mathcal{M} \cup F]$.

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.

$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$

Let $F \subseteq \mathcal{J}$ be the set of fractionally assigned jobs.
Let $H := G[\mathcal{M} \cup F]$.

Observe: (i, j) is an edge in $H \Leftrightarrow 0 < x_{ij} < 1$

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$
with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.
$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$

Let $F \subseteq \mathcal{J}$ be the set of fractionally assigned jobs.
Let $H := G[\mathcal{M} \cup F]$.

Observe: (i, j) is an edge in $H \Leftrightarrow 0 < x_{ij} < 1$

A matching in H is called F-*perfect* if it matches every vertex in F.

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.
$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$

Let $F \subseteq \mathcal{J}$ be the set of fractionally assigned jobs.
Let $H := G[\mathcal{M} \cup F]$.

Observe: (i, j) is an edge in $H \Leftrightarrow 0 < x_{ij} < 1$

A matching in H is called F-*perfect* if it matches every vertex in F.

Main step: Show that H always has an F-perfect matching.

Extreme Point Solutions of LP(T)

Definition: Bipartite Graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ with $(i, j) \in E \Leftrightarrow x_{ij} \neq 0$.

Jobs can be assigned *integrally* or *fractionally*.
$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$

Let $F \subseteq \mathcal{J}$ be the set of fractionally assigned jobs.
Let $H := G[\mathcal{M} \cup F]$.

Observe: (i, j) is an edge in $H \Leftrightarrow 0 < x_{ij} < 1$

A matching in H is called F-*perfect* if it matches every vertex in F.

Main step: Show that H always has an F-perfect matching.

Why is that useful . . . ?

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}.

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}. Let τ be the makespan of this schedule.

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}. Let τ be the makespan of this schedule.

By a binary search in the interval $[\frac{\tau}{|\mathcal{M}|}, \tau]$, find the smallest value of $T \in \mathbb{Z}^+$ for which LP(T) has a feasible solution. Let T^* be this value.

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}. Let τ be the makespan of this schedule.

By a binary search in the interval $[\frac{\tau}{|\mathcal{M}|}, \tau]$, find the smallest value of $T \in \mathbb{Z}^+$ for which LP(T) has a feasible solution. Let T^* be this value.

Find an extreme point solution x for LP(T^*).

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}. Let τ be the makespan of this schedule.

By a binary search in the interval $[\frac{\tau}{|\mathcal{M}|}, \tau]$, find the smallest value of $T \in \mathbb{Z}^+$ for which LP(T) has a feasible solution. Let T^* be this value.

Find an extreme point solution x for LP(T^*).

Assign all integrally set jobs to machines as in x.

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}. Let τ be the makespan of this schedule.

By a binary search in the interval $[\frac{\tau}{|\mathcal{M}|}, \tau]$, find the smallest value of $T \in \mathbb{Z}^+$ for which LP(T) has a feasible solution. Let T^* be this value.

Find an extreme point solution x for LP(T^*).

Assign all integrally set jobs to machines as in x.

Construct the graph H and find an F-perfect matching P in it (see Lemma 4 later, F is set of fractionally assg. jobs)

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}. Let τ be the makespan of this schedule.

By a binary search in the interval $[\frac{\tau}{|\mathcal{M}|}, \tau]$, find the smallest value of $T \in \mathbb{Z}^+$ for which LP(T) has a feasible solution. Let T^* be this value.

Find an extreme point solution x for LP(T^*).

Assign all integrally set jobs to machines as in x.

Construct the graph H and find an F-perfect matching P in it (see Lemma 4 later, F is set of fractionally assg. jobs)

Assign the fractional jobs to machines using P.

Algorithm

Assign job J_j to machine M_i that minimizes p_{ij}. Let τ be the makespan of this schedule.

By a binary search in the interval $[\frac{\tau}{|\mathcal{M}|}, \tau]$, find the smallest value of $T \in \mathbb{Z}^+$ for which LP(T) has a feasible solution. Let T^* be this value.

Find an extreme point solution x for LP(T^*).

Assign all integrally set jobs to machines as in x.

Construct the graph H and find an F-perfect matching P in it (see Lemma 4 later, F is set of fractionally assg. jobs)

Assign the fractional jobs to machines using P.

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Approximation Factor

$$\sum_{(i,j) \in S_{T^*}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_{T^*}} x_{ij} p_{ij} \leq T^*, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S$$

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Proof. $T^* \leq \text{OPT}$

Approximation Factor

$$\sum_{(i,j) \in S_{T^*}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_{T^*}} x_{ij} p_{ij} \leq T^*, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S$$

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Proof. $T^* \leq \text{OPT}$

Let x be an extreme point solution for $LP(T^*)$

Approximation Factor

$$\sum_{(i,j) \in S_{T^*}} x_{ij} = 1, \quad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_{T^*}} x_{ij} p_{ij} \leq T^*, \quad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \quad (i,j) \in S$$

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Proof. $T^* \leq \text{OPT}$

Let x be an extreme point solution for $LP(T^*)$

Fractional solution: makespan $\leq T^*$.

Approximation Factor

$$\sum_{(i,j) \in S_{T^*}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_{T^*}} x_{ij} p_{ij} \leq T^*, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S$$

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Proof. $T^* \leq \mathrm{OPT}$

Let x be an extreme point solution for $LP(T^*)$

Fractional solution: makespan $\leq T^*$.

$\Rightarrow$ Restriction to integral jobs has makespan $\leq T^*$.

Approximation Factor

$$\sum_{(i,j) \in S_{T^*}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j) \in S_{T^*}} x_{ij} p_{ij} \leq T^*, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S$$

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Proof. $T^* \leq \mathrm{OPT}$

Let x be an extreme point solution for $LP(T^*)$

Fractional solution: makespan $\leq T^*$.

$\Rightarrow$ Restriction to integral jobs has makespan $\leq T^*$.

For each edge $(i,j) \in S_{T^*}$: $p_{ij} \leq T^*$

Approximation Factor

$$\sum_{(i,j)\in S_{T^*}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_{T^*}} x_{ij} p_{ij} \le T^*, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \ge 0, \qquad (i,j) \in S$$

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Proof. $T^* \le \text{OPT}$

Let x be an extreme point solution for $LP(T^*)$

Fractional solution: makespan $\le T^*$.

$\Rightarrow$ Restriction to integral jobs has makespan $\le T^*$.

For each edge $(i,j) \in S_{T^*}$: $p_{ij} \le T^*$

Matching: ≤ 1 extra jobs per maschine

Approximation Factor

$$\sum_{(i,j)\in S_{T^*}} x_{ij} = 1, \qquad J_j \in \mathcal{J}$$

$$\sum_{(i,j)\in S_{T^*}} x_{ij} p_{ij} \leq T^*, \qquad M_i \in \mathcal{M}$$

$$x_{ij} \geq 0, \qquad (i,j) \in S$$

Theorem. This algorithm is a factor-2-approximation (assuming that we have an F-perfect matching).

Proof. $T^* \leq \mathrm{OPT}$

Let x be an extreme point solution for $LP(T^*)$

Fractional solution: makespan $\leq T^*$.

$\Rightarrow$ Restriction to integral jobs has makespan $\leq T^*$.

For each edge $(i,j) \in S_{T^*}$: $p_{ij} \leq T^*$

Matching: ≤ 1 extra jobs per maschine

$\Rightarrow$ total makespan $\leq 2T^* \leq 2\mathrm{OPT}$ $\square$

Approximation Algorithms

Lecture 7:
Scheduling Jobs on Parallel Machines

Part IV:
Pseudo-Trees and -Forests

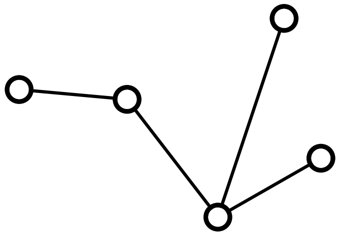
Joachim Spoerhase

Winter 2020/21

Pseudo-Trees and -Forests

Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

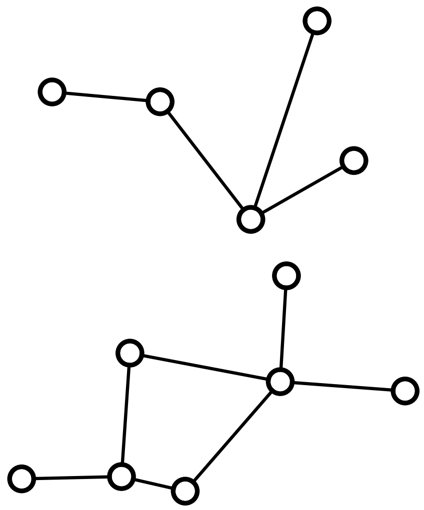
Pseudo-Trees and -Forests



Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree

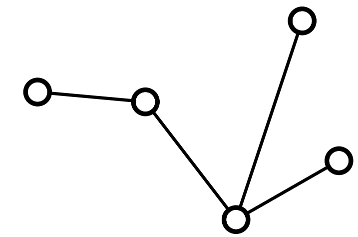
Pseudo-Trees and -Forests



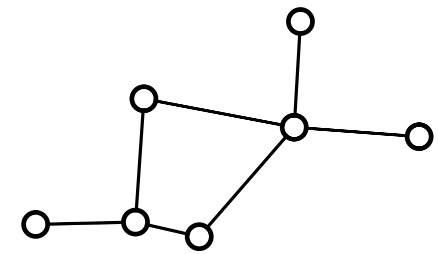
Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Trees and -Forests



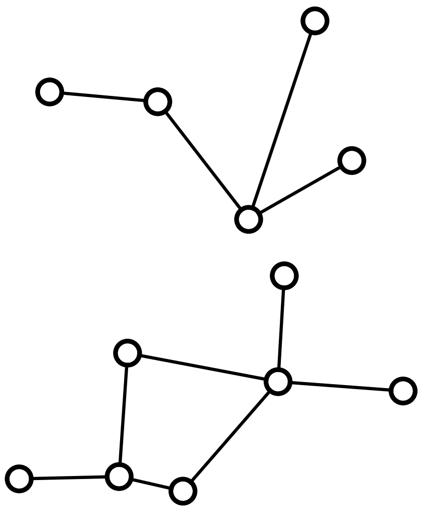
Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.



A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Forest: Collection of disjoint pseudo-trees.

Pseudo-Trees and -Forests



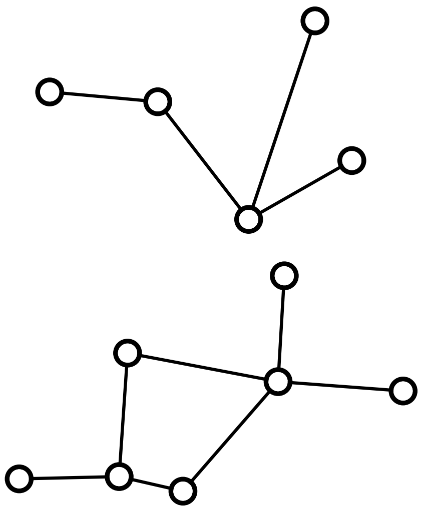
Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Forest: Collection of disjoint pseudo-trees.

Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

Pseudo-Trees and -Forests



Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

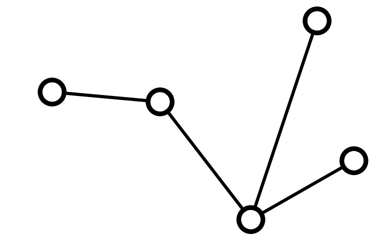
A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Forest: Collection of disjoint pseudo-trees.

Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

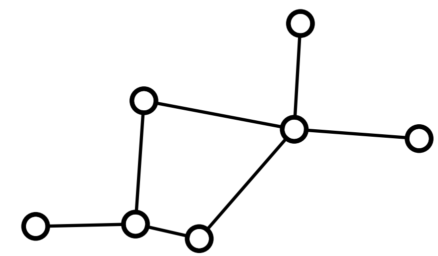
Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).

Pseudo-Trees and -Forests



Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree or a tree plus a single edge.



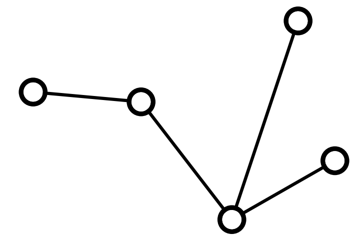
Pseudo-Forest: Collection of disjoint pseudo-trees.

Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

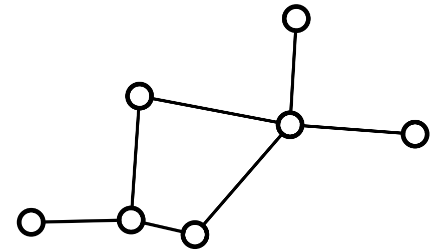
Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).

Each component of G corresponds to an extreme point solution.

Pseudo-Trees and -Forests



Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.



A pseudo-tree is either a tree or a tree plus a single edge.

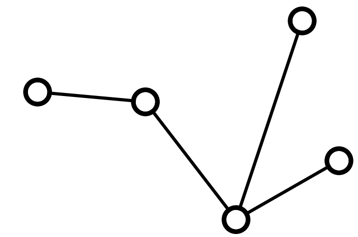
Pseudo-Forest: Collection of disjoint pseudo-trees.

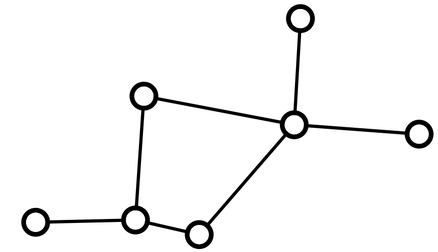
Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).
Each component of G corresponds to an extreme point solution.

Lemma 4.
The graph H has an F-perfect matching.

Pseudo-Trees and -Forests





Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Forest: Collection of disjoint pseudo-trees.

Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

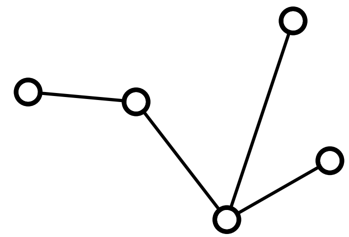
Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).

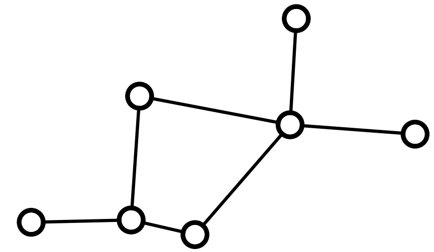
Each component of G corresponds to an extreme point solution.

Lemma 4.
The graph H has an F-perfect matching.

H is also a pseudo-forest: remove 1 edge per $v \in \mathcal{J} \setminus F$

Pseudo-Trees and -Forests





Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Forest: Collection of disjoint pseudo-trees.

Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

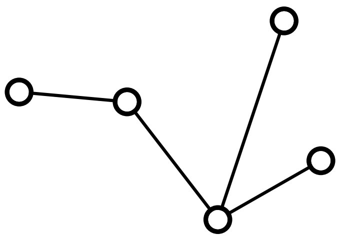
Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).

Each component of G corresponds to an extreme point solution.

Lemma 4.
The graph H has an F-perfect matching.

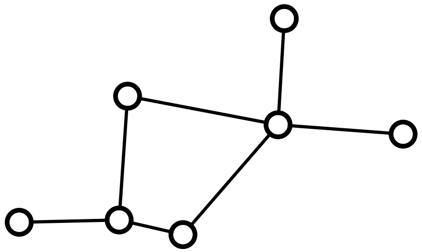
H is also a pseudo-forest: remove 1 edge per $v \in \mathcal{J} \setminus F$
Vertices in F have min. degree 2.

Pseudo-Trees and -Forests



Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree or a tree plus a single edge.



Pseudo-Forest: Collection of disjoint pseudo-trees.

Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

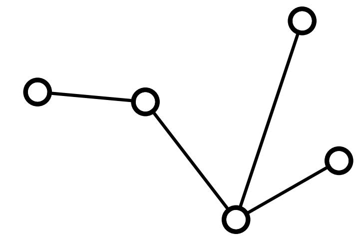
Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).

Each component of G corresponds to an extreme point solution.

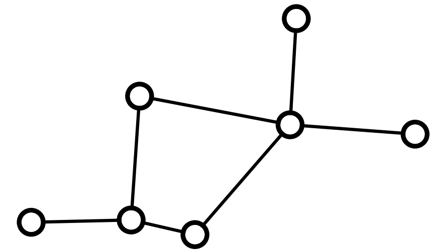
Lemma 4.
The graph H has an F-perfect matching.

H is also a pseudo-forest: remove 1 edge per $v \in \mathcal{J} \setminus F$
Vertices in F have min. degree 2. $\Rightarrow$ The leaves in H are machines.

Pseudo-Trees and -Forests



Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.



A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Forest: Collection of disjoint pseudo-trees.

> **Lemma 3.**
> The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).

Each component of G corresponds to an extreme point solution.

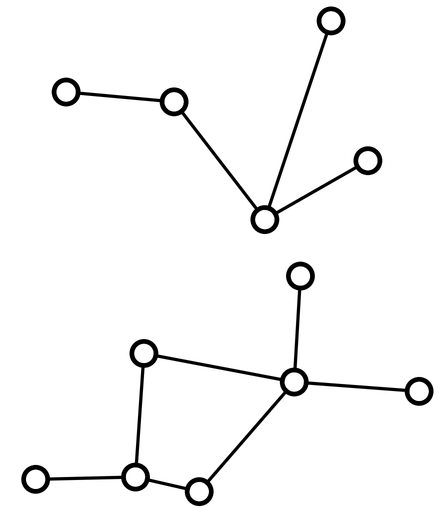
> **Lemma 4.**
> The graph H has an F-perfect matching.

H is also a pseudo-forest: remove 1 edge per $v \in \mathcal{J} \setminus F$

Vertices in F have min. degree 2. $\Rightarrow$ The leaves in H are machines.

After iteratively picking all leaves,

Pseudo-Trees and -Forests



Pseudo-Tree: A connected graph $G = (V, E)$ with at most $|V|$ edges.

A pseudo-tree is either a tree or a tree plus a single edge.

Pseudo-Forest: Collection of disjoint pseudo-trees.

Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

Extreme point solutions have $\leq |\mathcal{M}| + |\mathcal{J}|$ variables > 0 (L1).

Each component of G corresponds to an extreme point solution.

Lemma 4.
The graph H has an F-perfect matching.

H is also a pseudo-forest: remove 1 edge per $v \in \mathcal{J} \setminus F$
Vertices in F have min. degree 2. $\Rightarrow$ The leaves in H are machines.
After iteratively picking all leaves, only even cycles remain.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight?

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Job J_1 jas processing time m on all machines,

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Job J_1 jas processing time m on all machines,

all other jobs have processing time 1 on each machine.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Job J_1 jas processing time m on all machines,

all other jobs have processing time 1 on each machine.

Optimum: one machine with J_1, and all others spread evenly.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Job J_1 jas processing time m on all machines,

all other jobs have processing time 1 on each machine.

Optimum: one machine with J_1, and all others spread evenly.

Algorithm:

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Job J_1 jas processing time m on all machines,

all other jobs have processing time 1 on each machine.

Optimum: one machine with J_1, and all others spread evenly.

Algorithm:

LP(T) has no feasible solutions for any $T < m$.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Job J_1 jas processing time m on all machines,

all other jobs have processing time 1 on each machine.

Optimum: one machine with J_1, and all others spread evenly.

Algorithm:

LP(T) has no feasible solutions for any $T < m$.

Extreme point solution: Assign $1/m$ of J_1 and $m - 1$ other jobs to each machine.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Tight? Yes!

Instance I_m:

m machines and $m^2 - m + 1$ jobs

Job J_1 jas processing time m on all machines,

all other jobs have processing time 1 on each machine.

Optimum: one machine with J_1, and all others spread evenly.

Algorithm:

LP(T) has no feasible solutions for any $T < m$.

Extreme point solution: Assign $1/m$ of J_1 and $m - 1$ other jobs to each machine.

$\Rightarrow$ Makespan $2m - 1$.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Better?

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Better?

No better approximation algorithm is known.

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Better?

No better approximation algorithm is known.

The problem cannot be approximated within factor $< 3/2$ (unless P=NP) [Lenstra, Shmoys & Tardos '90]

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Better?

No better approximation algorithm is known.

The problem cannot be approximated within factor $< 3/2$ (unless P=NP) [Lenstra, Shmoys & Tardos '90]

For a constant number of machines, for every $\varepsilon > 0$ there is a factor-$(1 + \varepsilon)$-approximation algorithm. [Horowitz & Sahni '76]

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Better?

No better approximation algorithm is known.

The problem cannot be approximated within factor $< 3/2$ (unless P=NP) [Lenstra, Shmoys & Tardos '90]

For a constant number of machines, for every $\varepsilon > 0$ there is a factor-$(1 + \varepsilon)$-approximation algorithm. [Horowitz & Sahni '76]

For uniform machines, for every $\varepsilon > 0$ there is a factor-$(1 + \varepsilon)$-approximation algorithm. [Hochbaum & Shmoys '87]

Scheduling on Parallel Machines

Theorem. There is an LP-based 2-approximation algorithm for the problem of scheduling jobs on unrelated parallel machines.

Better?

No better approximation algorithm is known.

The problem cannot be approximated within factor $< 3/2$ (unless P=NP) [Lenstra, Shmoys & Tardos '90]

For a constant number of machines, for every $\varepsilon > 0$ there is a factor-$(1 + \varepsilon)$-approximation algorithm. [Horowitz & Sahni '76]

For uniform machines, for every $\varepsilon > 0$ there is a factor-$(1 + \varepsilon)$-approximation algorithm. [Hochbaum & Shmoys '87]
(Machines have different speed, but process jobs uniformly.)