Approximation Algorithms

Lecture 7:
Scheduling Jobs on Parallel Machines

Part I:
ILP & Parametric Pruning

Philipp Kindermann   Summer Semester 2020

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

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

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

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

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	p_{11}	p_{13}	p_{18}

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.

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

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

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

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** $\sigma: \mathcal{J} \to \mathcal{M}$ 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 | p_{11} | p_{13} | p_{18}

M_2 | p_{22} | p_{27}

M_3 | p_{34} | p_{35} | p_{36}

$$\mathcal{J} = \{J_1, J_2, \dots, J_8\}$$
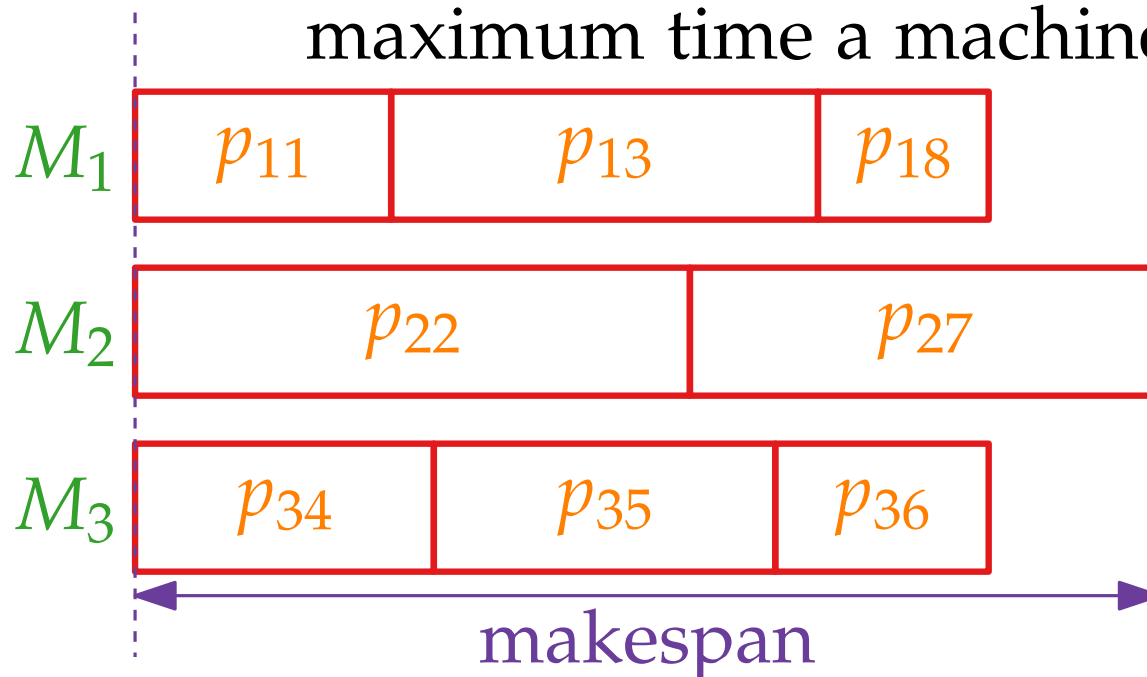
$$\mathcal{M} = \{M_1, M_2, M_3\}$$

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

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.

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

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

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

Formulation as ILP

> **minimize** t
>
> **subject to**

Formulation as ILP

minimize $\quad t$

subject to

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

Formulation as ILP

minimize $\quad t$

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

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

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

Formulation as ILP

$$\textbf{minimize} \quad t$$

$$\textbf{subject to} \quad \sum_{M_i \in \mathcal{M}} x_{ij} = 1, \quad J_j \in \mathcal{J}$$

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

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

Formulation as ILP

$$\textbf{minimize} \quad t$$

$$\textbf{subject to} \quad \sum_{M_i \in \mathcal{M}} x_{ij} = 1, \quad J_j \in \mathcal{J}$$

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

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

Formulation as ILP

$$\textbf{minimize} \quad t$$

$$\textbf{subject to} \quad \sum_{M_i \in \mathcal{M}} x_{ij} = 1, \quad J_j \in \mathcal{J}$$

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

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

Task: Prove that the integrality gap is unbounded!

Formulation as ILP

$$\textbf{minimize} \quad t$$

$$\textbf{subject to} \quad \sum_{M_i \in \mathcal{M}} x_{ij} = 1, \quad J_j \in \mathcal{J}$$

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

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

Task: Prove that the integrality gap is unbounded!

Solution: m machines and one job with processing time m

Formulation as ILP

$$\textbf{minimize} \quad t$$

$$\textbf{subject to} \quad \sum_{M_i \in \mathcal{M}} x_{ij} = 1, \quad J_j \in \mathcal{J}$$

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

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

Task: Prove that the integrality gap is unbounded!

Solution: m machines and one job with processing time m

$\Rightarrow \text{OPT} = m$ 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):

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

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

Define the "pruned" relaxation LP(T):

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

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

Define the "pruned" relaxation LP(T):

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

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 $\text{LP}(T)$:

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

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):

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

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

Define the "pruned" relaxation LP(T):

minimize t

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

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

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

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):

$$
\begin{aligned}
&\textbf{minimize} \quad t \\[2mm]
&\textbf{subject to} \quad \sum_{(i,j) \in S_T,\ M_i \in \mathcal{M}} x_{ij} = 1, \quad J_j \in \mathcal{J} \\[2mm]
&\qquad\qquad\quad \sum_{(i,j) \in S_T,\ J_j \in \mathcal{J}} x_{ij} p_{ij} \leq t,\ T \quad M_i \in \mathcal{M} \\[2mm]
&\qquad\qquad\quad x_{ij} \in \{0,1\},\ \geq 0 \quad M_i \in \mathcal{M}, J_j \in \mathcal{J} \quad (i,j) \in S_T
\end{aligned}
$$

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

Define the "pruned" relaxation LP(T):

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

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):

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

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

Paremetric Pruning

Strengthen the ILP $\rightarrow$ 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):

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

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

Philipp Kindermann   Summer Semester 2020

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that $\mathrm{LP}(T)$ has a solution.

$$\mathrm{LP}(T)$$

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

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

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

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.

$$\text{LP}(T)$$

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

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

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

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?

$$\text{LP}(T)$$

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

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

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

Properties of Extreme Point Solutions

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

What are the bounds for our search?

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

$$\text{LP}(T)$$

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

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

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

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: $T^* \leq$ OPT

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

$$\text{LP}(T)$$

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

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

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

Properties of Extreme Point Solutions

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

What are the bounds for our search?

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

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

$$\text{LP}(T)$$

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

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

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

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

Properties of Extreme Point Solutions

Use binary search to find the smallest T so that $\text{LP}(T)$ 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 $\text{LP}(T^*)$ to a schedule whose makespan is $\leq 2T^*$

$$\text{LP}(T)$$

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

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

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

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

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

Lemma 1

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

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

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

Lemma 1.

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

Lemma 1

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

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

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

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

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

Lemma 1

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

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

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

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

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

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 $\text{LP}(T)$ has $\leq |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

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

Lemma 1

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

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

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

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

Proof. $L(T)$: $|S_T|$ variables
extreme point sol.: $|S_T|$ inequalities exactly fulfilled
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 $\mathrm{LP}(T)$ has $\leq |\mathcal{M}| + |\mathcal{J}|$ pos. variables.

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

extreme point sol.: $|S_T|$ inequalities exactly fulfilled

max. $|\mathcal{J}|$

max. $|\mathcal{M}|$

$\Rightarrow$ min. $|S_T| - |\mathcal{J}| - |\mathcal{M}|$

Lemma 1

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

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

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

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

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

max. $|\mathcal{J}|$

max. $|\mathcal{M}|$

$\Rightarrow$ min. $|S_T| - |\mathcal{J}| - |\mathcal{M}|$

$\Rightarrow$ max. $|\mathcal{M}| + |\mathcal{J}|$ not fulfilled $\qquad \square$

Lemma 2

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

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

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

Lemma 2.

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

Lemma 2

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

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

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

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

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

Lemma 2

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

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

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

Lemma 2.
Any extreme point solution for $\mathrm{LP}(T)$ 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.

Lemma 2

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

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

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

Lemma 2.
Any extreme point solution for $\mathrm{LP}(T)$ 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

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

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

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

Lemma 2.
Any extreme point solution for $\text{LP}(T)$ 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

Lemma 2

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

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

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

Lemma 2.
Any extreme point solution for $\mathrm{LP}(T)$ 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

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

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

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

Lemma 2.
Any extreme point solution for $\mathrm{LP}(T)$ 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)

Lemma 2

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

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

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

Lemma 2.
Any extreme point solution for $\mathrm{LP}(T)$ 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

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

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

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

Lemma 2.
Any extreme point solution for $\text{LP}(T)$ 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)
$\Rightarrow \beta \leq |\mathcal{M}| \quad \Rightarrow \alpha \geq |\mathcal{J}| - |\mathcal{M}|$ $\qquad \square$

Approximation Algorithms

Lecture 7:
Scheduling Jobs on Parallel Machines

Part III:
An Algorithm

Philipp Kindermann   Summer Semester 2020

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 *integral* or *fractional*.

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 *integral* or *fractional*.

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

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 *integral* or *fractional*.
$$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$$

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 *integral* or *fractional*.
$$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$$

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 *integral* or *fractional*.
$$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$$

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 *integral* or *fractional*.
$$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$$

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 *integral* or *fractional*.
$$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$$

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 *integral* or *fractional*.
$$(\exists M_i \in \mathcal{M} : 0 < x_{ij} < 1)$$

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

Find an extreme point solution x for $\text{LP}(T^*)$.

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 $\left[\frac{\tau}{|\mathcal{M}|}, \tau\right]$, 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 $\left[\frac{\tau}{|\mathcal{M}|}, \tau\right]$, 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 a perfect matching P in it (see Lemma 4 later).

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 $\left[\frac{\tau}{|\mathcal{M}|}, \tau\right]$, 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 a perfect matching P in it (see Lemma 4 later).

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

Find an extreme point solution x for $\mathrm{LP}(T^*)$.

Assign all integrally set jobs to machines as in x.

Construct the graph H and find a perfect matching P in it (see Lemma 4 later).

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

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

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

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

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

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

Approximation Factor

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

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

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

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

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

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

Approximation Factor

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

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

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

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 $LP(T^*)$

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

Approximation Factor

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

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

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

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 $\le T^*$.

Approximation Factor

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

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

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

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 $LP(T^*)$

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

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

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

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

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

Proof. $T^* \leq$ 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

Approximation Factor

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

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

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

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

Proof. $T^* \leq$ 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 $\leq 2T^* \leq 2$OPT $\square$

Approximation Algorithms

Lecture 7:
Scheduling Jobs on Parallel Machines

Part IV:
Pseudo-Trees and -Forests

Philipp Kindermann   Summer Semester 2020

Pseudo-Trees and -Forests

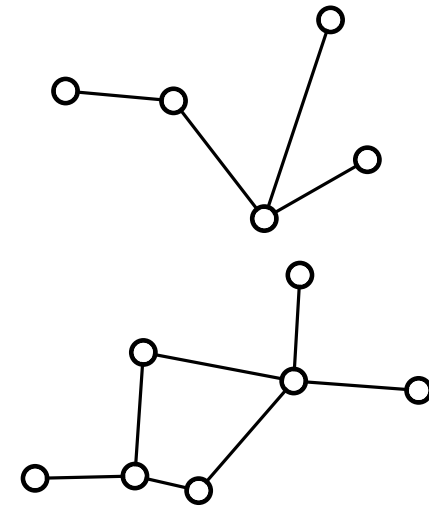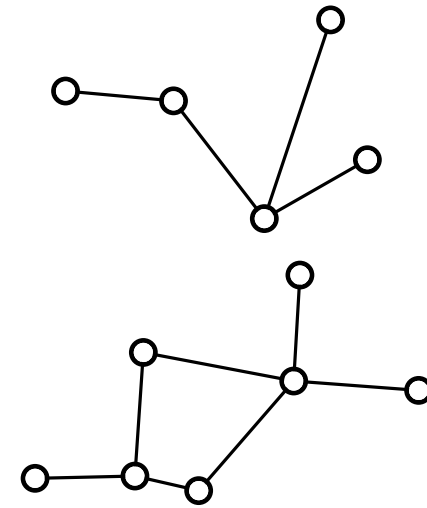**Pseudo-Tree**:   A connected graph $G = (V, E)$ with at most $|V|$ edges.

Pseudo-Trees and -Forests

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

A pseudo-tree is either a tree
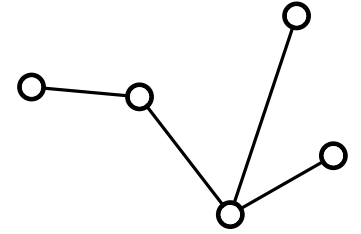
Pseudo-Trees and -Forests

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.
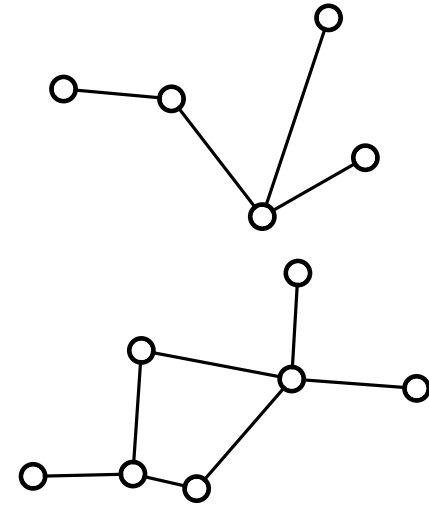
Pseudo-Trees and -Forests

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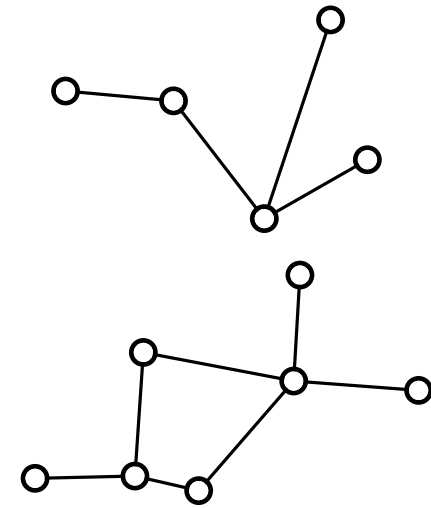The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

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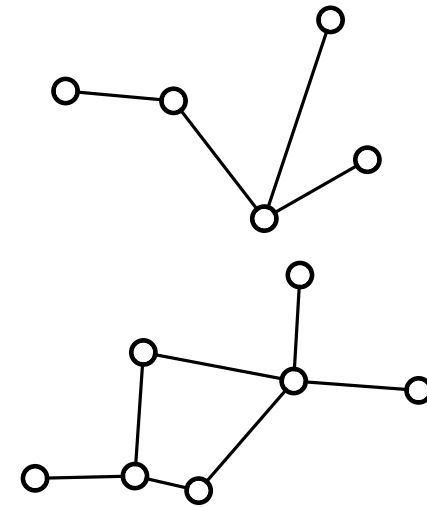
Lemma 3.
The bipartite graph $G = (\mathcal{M} \cup \mathcal{J}, E)$ is a pseudo-forest.

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).
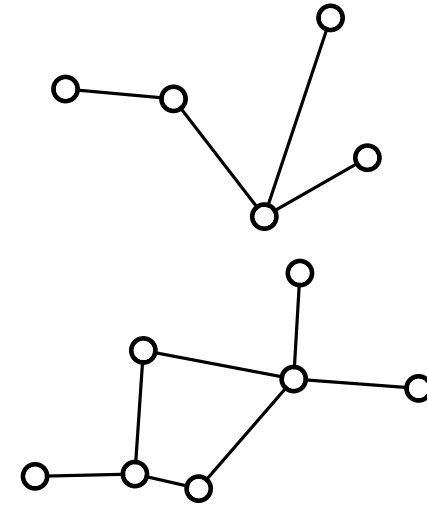Each component of G corresponds to an extreme point solution.

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).
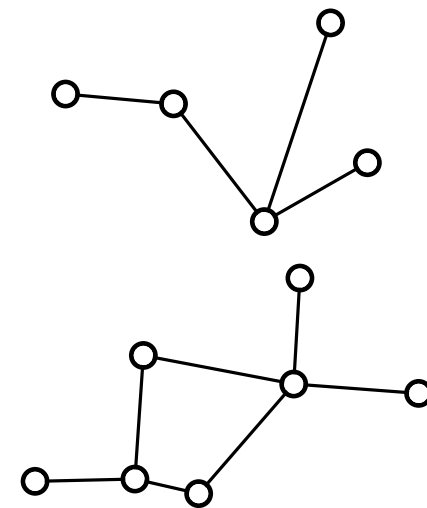Each component of G corresponds to an extreme point solution.

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.
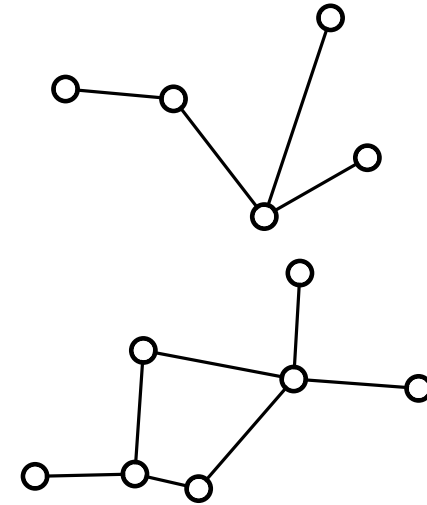The graph H has an F-perfect matching.

H ist 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.

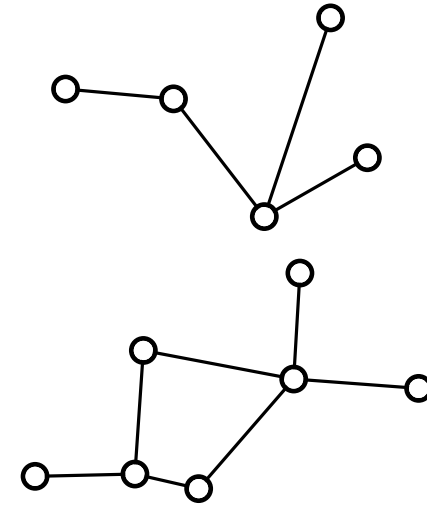**Lemma 4.**
The graph H has an F-perfect matching.

H ist 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.

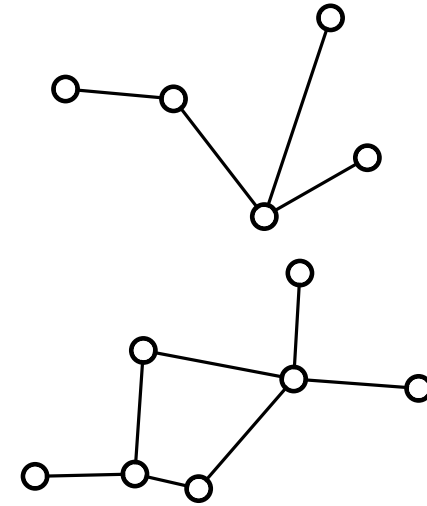H ist also a pseudo-forest: remove 1 edge per $v \in \mathcal{J} \setminus F$
Vertices in F have min. degree 2. $\Rightarrow$ The leaves in H are machines.

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 ist 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.
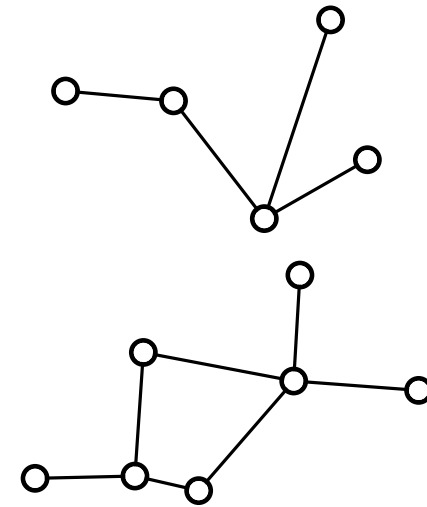After iteratively picking all leaves,

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 ist 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.)