

Approximation Algorithms

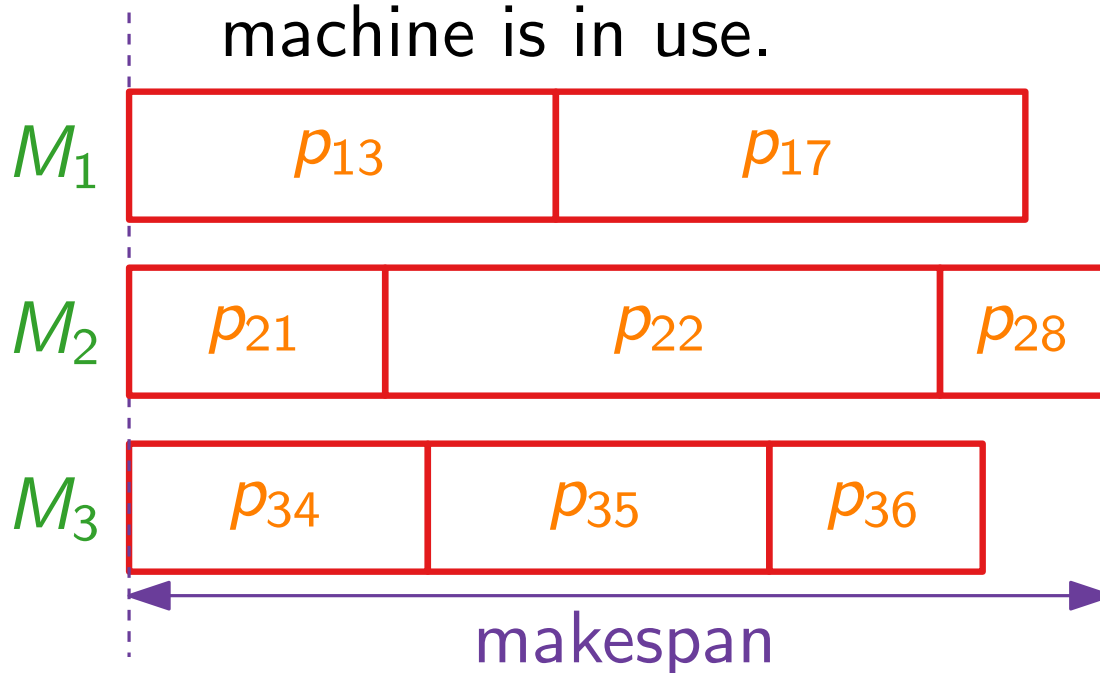
Lecture 7: Scheduling Jobs on Parallel Machines

Part I: ILP & Parametric Pruning

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



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

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

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

Formulation as ILP

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

Task: Prove that the integrality gap is unbounded!

Solution: m machines and one job with processing time m
 $\Rightarrow \text{OPT} = m$ and $\text{OPT}_{\text{frac}} = 1$.

Parametric Pruning

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

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

Introduce new parameter $T \in \mathbb{N}$ as 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 $\text{LP}(T)$:

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

Note:

$\text{LP}(T)$ has no objective function; we just need to check whether 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

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 at most $2T^*$.

$\text{LP}(T)$:

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

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

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

Lemma 1.

Every extreme-point solution of $\text{LP}(T)$ has at most $|\mathcal{J}| + |\mathcal{M}|$ positive variables.

Lemma 2.

Every extreme-point solution of $\text{LP}(T)$ sets at least $|\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Lemma 1

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

Lemma 1.

Every extreme-point solution of $\text{LP}(\mathcal{T})$ has at most $|\mathcal{J}| + |\mathcal{M}|$ positive variables.

Proof.

$\text{LP}(\mathcal{T})$: $|S_T|$ variables

extreme-point solution: $|S_T|$ inequalities tight

→ at most $|\mathcal{J}|$ inequalities

→ at most $|\mathcal{M}|$ inequalities

⇒ At least $|S_T| - |\mathcal{J}| - |\mathcal{M}|$ variables are 0.

⇒ At most $|\mathcal{J}| + |\mathcal{M}|$ variables are positive. $\square$

Lemma 2

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

Lemma 2.

Every extreme-point solution of $\text{LP}(T)$ sets at least $|\mathcal{J}| - |\mathcal{M}|$ jobs integrally.

Proof. Let x be an extreme-point solution of $\text{LP}(T)$.
 Assume x has α integral jobs and β fractional jobs.
 $\Rightarrow \alpha + \beta = |\mathcal{J}|$
 Each fractional job runs on at least two machines.
 $\Rightarrow$ For each such job, at least two variables are pos.
 $\Rightarrow \alpha + 2\beta \leq |\mathcal{J}| + |\mathcal{M}|$ (Lemma 1)
 $\Rightarrow \beta \leq |\mathcal{M}|$ and $\alpha \geq |\mathcal{J}| - |\mathcal{M}|$ □

Evaluation of the Lecture and the Tutorial

- Please participate **now**! Check your email.
- It takes only a few minutes.
- We get the results only if at least 6 students participate.
- Please evaluate Alexander Wolff (for the lecture) and Antonio Lauerbach (for the tutorial).

Approximation Algorithms

Lecture 7: Scheduling Jobs on Parallel Machines

Part III: An Algorithm

Extreme Point Solutions of $LP(T)$

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

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.

And why is this useful ...?

Algorithm

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

Let τ be the makespan of this schedule.

Do a binary search in the interval $[\frac{\tau}{|\mathcal{M}|}, \tau]$ to find the smallest value T^* of $T \in \mathbb{Z}^+$ s.t. $\text{LP}(T)$ has a feasible solution.

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

Assign all integrally set jobs to machines as in x .

Construct the graph H and find an F -perfect matching P in H , where F is the set of fractionally assigned jobs ($\rightarrow$ Lem. 4 later)

Assign the fractional jobs to machines using P .

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

Approximation Factor

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

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

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

Let x be an extreme-point solution for $\text{LP}(T^*)$

→ Fractional solution: Makespan $\leq T^*$.

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

For each edge $(i,j) \in S_{T^*}$, it holds that $p_{ij} \leq T^*$. Pruning!

Matching: at most one extra job per machine.

⇒ total makespan $\leq 2T^* \leq 2\text{OPT}$

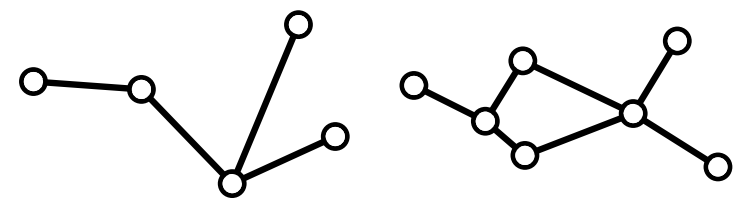
□

Approximation Algorithms

Lecture 7: Scheduling Jobs on Parallel Machines

Part IV: Pseudo-Trees and -Forests

Pseudo-Trees and -Forests



Pseudo-tree: a connected graph with at most as many edges as vertices.
(A pseudo-tree is either a tree or a tree plus a single edge.)

Pseudo-forest: a 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 $\leq |\mathcal{M}| + |\mathcal{J}|$ positive variables (Lemma 1).
Each conn. component C of G corresponds to an extreme-point solution.
(Suppose not. Then the solution that corresponds to C is the convex combination of other solutions. But this contradicts the definition of G .)
 $\Rightarrow C$ has at most as many edges (pos. var.) as vertices (jobs+machines).

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

In G , every vertex in $\mathcal{J} \setminus F$ is a leaf. $\xRightarrow{\text{remove leaves}}$ H is a pseudo-forest, too.
Vertices in F have minimum degree 2. $\Rightarrow$ The leaves in H are machines.
After iteratively matching all leaves, only *even* cycles remain. (H is bipartite :-)

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 has processing time m on every machine,
all other jobs have processing time 1 on every machine.

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

Algorithm: $\Rightarrow$ Makespan = m .

LP(T) has no feasible solution 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.

Can we do 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 may have different speeds, but process jobs uniformly.)