Introduction to
Approximation Algorithms

Mong-Jen Kao (高孟駿)

Friday 13:20 – 15:10

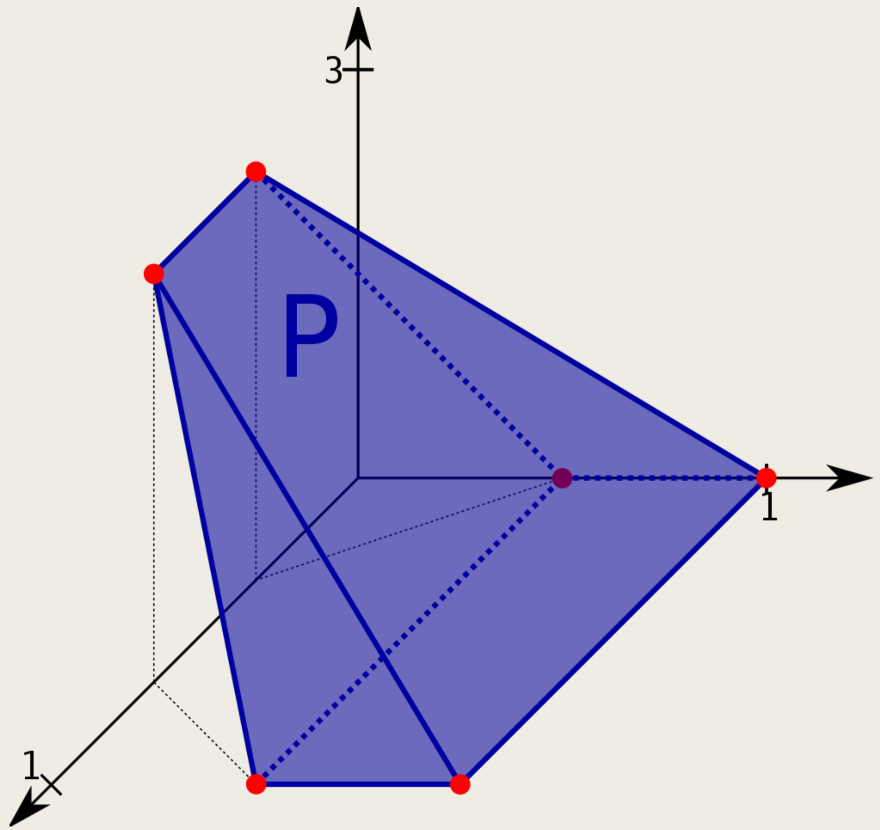
Outline

- Extreme Point Structure of LPs
- Half-Integrality of Vertex Cover
- Unrelated Machine Scheduling
  - A strengthened LP (*) and parametric search
  - Extreme point structure of (*)
  - A 2-approximation algorithm

Extreme Point Structure of LPs

Extreme Points of a Polytope

- Consider the convex polytope Q defined by $A\boldsymbol{x} \leq b$, where $x \in \mathbb{R}^n$.



Extreme Points of a Polytope

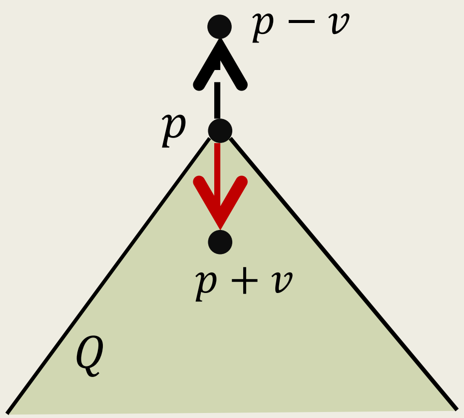
- Consider the convex polytope Q defined by $A\boldsymbol{x} \leq b$, where $x \in \mathbb{R}^n$.

Definition. (Extreme Point)

A point $p \in Q$ is an *extreme point* if for any (vector) $v \in \mathbb{R}^n$,
$p + v \in Q$ implies that $p - v \notin Q$.

- Such a point is also called a ***vertex*** of Q, or, a ***basic feasible*** solution for $A\boldsymbol{x} \leq b$.

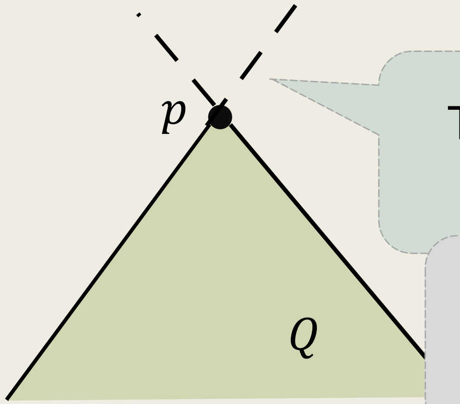
- An equivalent definition is that, $p \in Q$ is an extreme point if $\nexists q, r \in Q$ such that $p = (q + r)/2$.



Equivalently,
$\nexists q, r \in Q,\ a \in \mathbb{R}$ such that $p = a \cdot q + (1 - a) \cdot r$.

Extreme Points Structure

- Let p be an extreme point for $A\boldsymbol{x} \leq b$, where $x \in \mathbb{R}^n$.
  - The point p lies in ***the hyperplanes*** *defined by some of the constraints in* $A\boldsymbol{x} \leq b$, with the inequality *holds with equality*.
  - Let $A'\boldsymbol{x} = b'$ be the system formed by these constraints, i.e., those in $A\boldsymbol{x} \leq b$ that hold with equality at p.
  - To *uniquely define* p, ***the matrix*** A' ***must have a rank of*** $\boldsymbol{n}$.



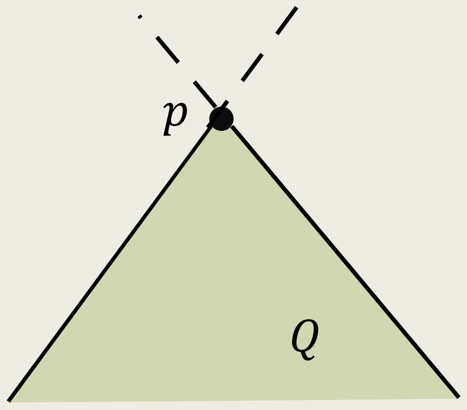
The set of ***hyperplanes that uniquely define*** p.

The set of **constraints in** $A\boldsymbol{x} \leq b$ **that hold with equality at** p.

For any extreme point p, there exists a set of $\boldsymbol{n}$ ***linearly independent constraints in*** $A\boldsymbol{x} \leq b$ ***that hold with equality at*** p.

Obtaining ***Optimal Extreme Point Solutions***

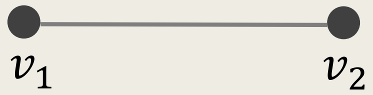
- Most LP solvers compute optimal extreme point solutions for the considered LP.
  - This includes the simplex method, interior-point method, and Ellipsoid method.
  - So, simply apply the solvers and you get an optimal extreme point solution for the LP.

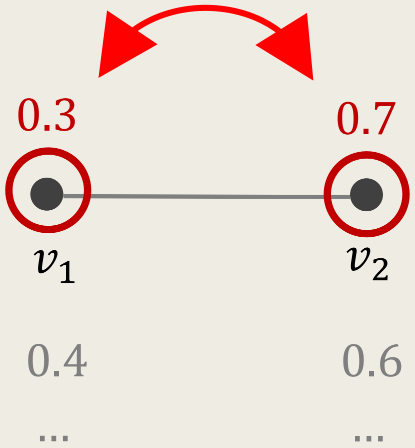


Why Extreme Point Solutions?

- Let's consider the *simple one-edge example* for vertex cover, and the *linear constraints* for it.

$$x_1 + x_2 \geq 1,$$
$$x_1, x_2 \geq 0.$$





  - For small $\epsilon > 0$,
    $(0.3 + \epsilon, 0.7 - \epsilon)$ and $(0.3 - \epsilon, 0.7 + \epsilon)$ are both feasible solutions.
    So, $(0.3, 0.7)$ is not extreme.
  - The only extreme point solutions are $(0,1)$ and $(1,0)$.

The extreme point solution moves the value ***greedily*** towards some direction.

The Half-Integrality of Vertex Cover

Half-Integrality of Vertex Cover

- Consider the natural LP relaxation for the vertex cover problem.

$$
\begin{aligned}
\min \quad & \sum_{v \in V} x_v && (*) \\
\text{s.t.} \quad & x_u + x_v \geq 1, && \forall\, (u, v) \in E, \\
& x_v \geq 0, && \forall\, v \in V.
\end{aligned}
$$

  - We will show that, any extreme point solution for (*) will set the value of each variable to be either 0, 1/2, or 1.
    - i.e., it will be ***half-integral***.

$$\min \sum_{v \in V} x_v \qquad (*)$$

$$\text{s.t.} \quad x_u + x_v \geq 1, \qquad \forall\, (u, v) \in E,$$

$$x_v \;\geq\; 0, \qquad \forall\, v \in V.$$

- Consider any feasible solution $\boldsymbol{x}$ for $(*)$ that is not half-integral,

  i.e., $\exists v \in V$ such that $x_v \notin \left\{0, \frac{1}{2}, 1\right\}$.

- We will show that $\boldsymbol{x}$ is not an extreme point solution.
  - The idea is to show that,

    $\exists\, \boldsymbol{p}$ such that, both $\boldsymbol{x} + \boldsymbol{p}$ and $\boldsymbol{x} - \boldsymbol{p}$ are feasible for LP$(*)$.
  - Let

$$V^+ = \left\{ v \in V : \frac{1}{2} < x_v < 1 \right\}, \text{and} \quad V^- = \left\{ v \in V : 0 < x_v < \frac{1}{2} \right\}$$

    be the set of large / small vertices that are not half-integrally set.

- Let

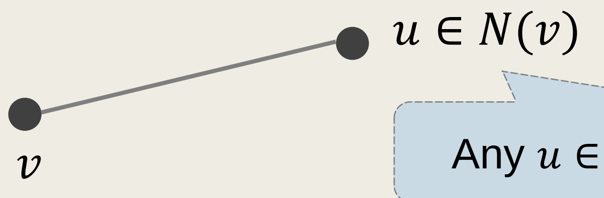
$$V^+ = \left\{ v \in V : \frac{1}{2} < x_v < 1 \right\}, \text{and} \quad V^- = \left\{ v \in V : 0 < x_v < \frac{1}{2} \right\}$$

be the set of large / small vertices that are not half-integrally set.

- Pick a ***sufficiently small*** $\epsilon > 0$, and define

$$y_v := \begin{cases} x_v + \epsilon, & \text{if } v \in V^+, \\ x_v - \epsilon, & \text{if } v \in V^-, \\ x_v, & \text{otherwise,} \end{cases} \qquad z_v := \begin{cases} x_v - \epsilon, & \text{if } v \in V^+, \\ x_v + \epsilon, & \text{if } v \in V^-, \\ x_v, & \text{otherwise.} \end{cases}$$

Intuitively, for any $v \in V^-$,



Any $u \in N(v)$ must belong to V^+.

Hence, the adjustment in y keeps the constraints satisfied, and y is feasible.

$$\begin{aligned} \min \quad & \sum_{v \in V} x_v & & (*) \\ \text{s.t.} \quad & x_u + x_v \geq 1, & & \forall\, (u,v) \in E, \\ & x_v \geq 0, & & \forall\, v \in V. \end{aligned}$$

- Let

$$V^+ = \left\{ v \in V : \frac{1}{2} < x_v < 1 \right\}, \text{and} \quad V^- = \left\{ v \in V : 0 < x_v < \frac{1}{2} \right\}$$

be the set of large / small vertices that are not half-integrally set.

- Pick a ***sufficiently small*** $\epsilon > 0$, and define

$$y_v := \begin{cases} x_v + \epsilon, & \text{if } v \in V^+, \\ x_v - \epsilon, & \text{if } v \in V^-, \\ x_v, & \text{otherwise}, \end{cases} \qquad z_v := \begin{cases} x_v - \epsilon, & \text{if } v \in V^+, \\ x_v + \epsilon, & \text{if } v \in V^-, \\ x_v, & \text{otherwise}. \end{cases}$$

Both $\boldsymbol{y}$ and $\boldsymbol{z}$ are feasible for $(*)$, and $\boldsymbol{x} = (\boldsymbol{y} + \boldsymbol{z})/2$.

Hence, $\boldsymbol{x}$ is not extreme for $(*)$.

$$\min \sum_{v \in V} x_v \qquad (*)$$

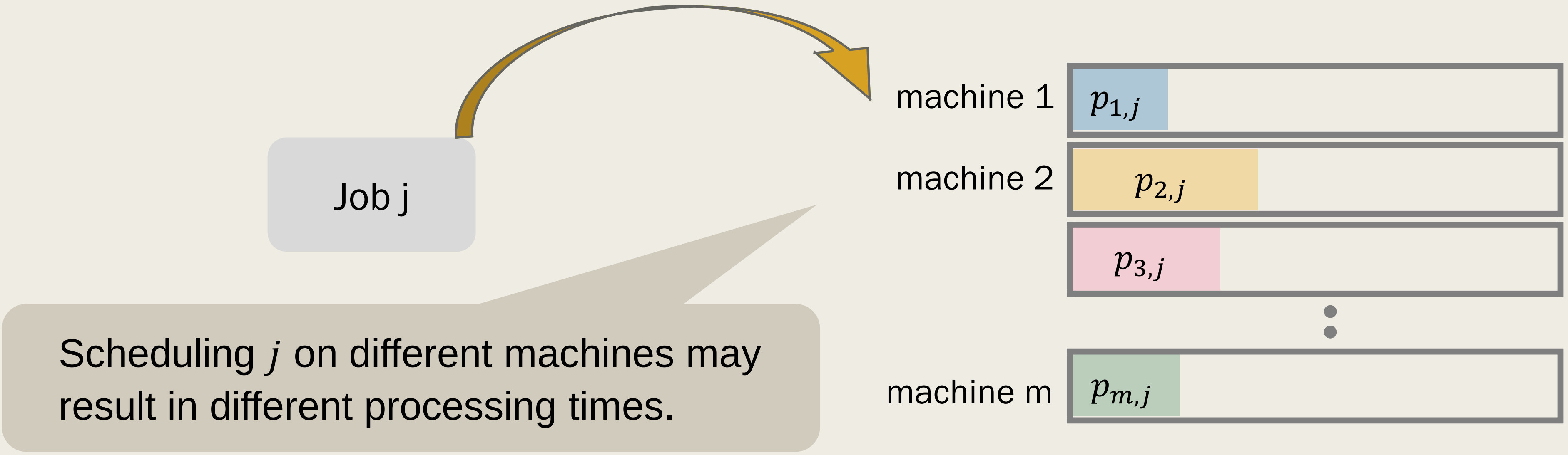
$$\text{s.t.} \quad x_u + x_v \geq 1, \qquad \forall\, (u, v) \in E,$$

$$x_v \geq 0, \qquad \forall\, v \in V.$$

Unrelated Machine Scheduling

Scheduling on Unrelated Parallel Machines

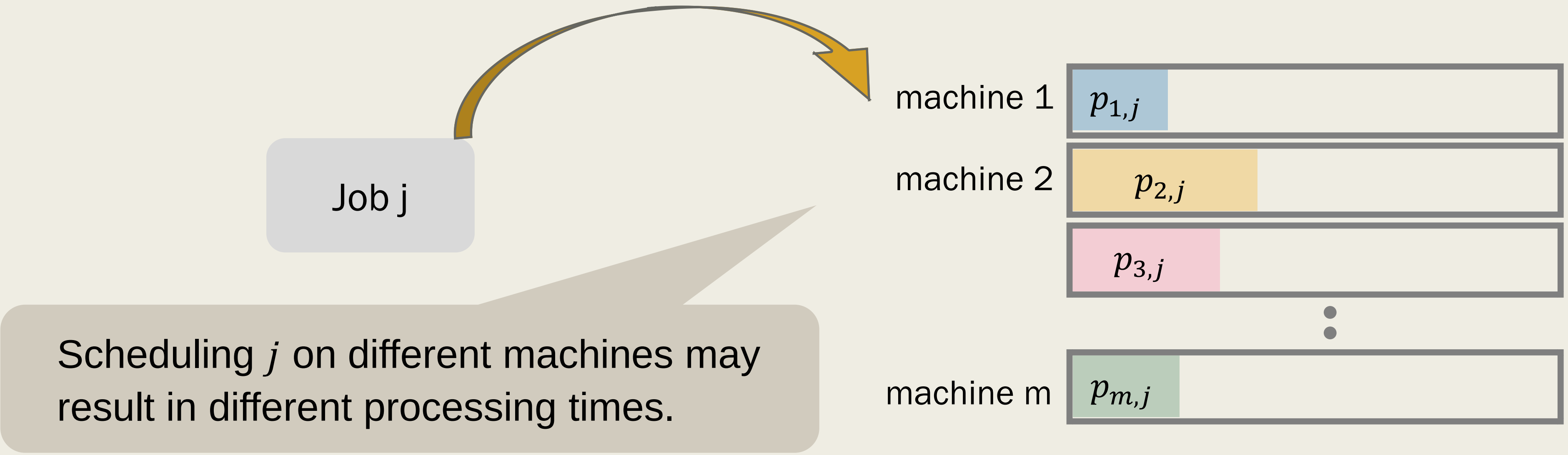
- Let J be a set of n jobs, M be a set of m machines, and $p_{i,j} \in \mathbb{Z}^+$ for each $j \in J, i \in M$ be the time it takes to process job j on machine i.
  - The



Scheduling on Unrelated Parallel Machines

- Given a set J of n jobs, a set M of machines, and for each $j \in J, i \in M$, $p_{i,j} \in \mathbb{Z}^+$ which is the time it takes to process job j on machine i,

  the goal of this problem is to schedule the jobs on the machines so as to ***minimize*** the ***maximum processing time of any machine***, i.e., to minimize the *makespan* of the schedule.



The Natural LP has an Unbounded Integrality Gap

- We can formulate the problem in the following natural way.
  - For each $i \in M, j \in J$, we have a variable $x_{i,j} \in \{0,1\}$.

    The constraints for feasibility of the schedule:

$$\sum_{i \in M} x_{i,j} = 1, \qquad \forall j \in J.$$

  - To model the objective value, we have a variable $t \in \mathbb{Z}^{\geq 0}$.

    The constraints for modeling the objective value:

$$\sum_{j \in J} p_{i,j} \cdot x_{i,j} \leq t, \qquad \forall i \in M.$$

- We obtain a natural LP for this problem.

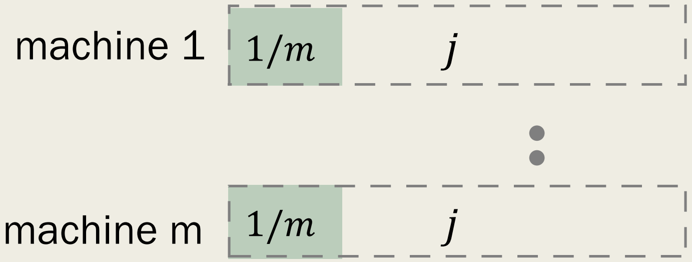
  However, this LP has an unbounded integrality gap.

  Consider the following example.

  Suppose that we have m machines and one job j with $p_{i,j} = m$ for all $1 \leq i \leq m$.

  The optimal fractional solution for (*) will set $x_{i,j} = 1/m$ for all $1 \leq i \leq m$, which results in a makespan of 1, while the optimal integral solution has a makespan of m.

$$
\begin{aligned}
\min \quad & t && (*) \\
\text{s.t.} \quad & \sum_{i \in M} x_{i,j} = 1, && \forall\, j \in J, \\
& \sum_{j \in J} p_{i,j} \cdot x_{i,j} \leq t, && \forall i \in M, \\
& t \geq 0, && \\
& x_{i,j} \;\geq\; 0, && \forall\, i \in M, j \in J.
\end{aligned}
$$



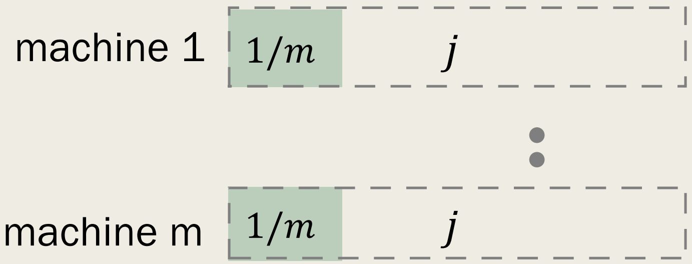
The problem is that, we allow jobs to be assigned to machines which has *strictly less completion time* than the job's processing time on that machine.

Can we strengthen the LP?

Consider the following example.

Suppose that we have m machines and one job j with $p_{i,j} = m$ for all $1 \leq i \leq m$.

The optimal fractional solution for (*) will set $x_{i,j} = 1/m$ for all $1 \leq i \leq m$, which results in a makespan of 1, while the optimal integral solution has a makespan of m.



The problem is that, we allow jobs to be assigned to machines which has <u>*strictly less completion time*</u> than the job's processing time on that machine.

The situation can be avoid, if we add the constraint to the relaxation :

$$\forall i \in M, j \in J: \quad \text{if } p_{i,j} > t, \qquad \text{then } \; x_{i,j} = 0.$$

However, *this is not a linear constraint*.

Parametric Search for Machine Scheduling

- In the following, we develop a parametric search process for this problem.
- Let t^* denote the optimal makespan and $T \in \mathbb{Z}^+$ be a guess for t^*.
  - Then, we know that, for any $T \geq t^*$, no assignments will be made between any $i \in M, j \in J$ with $p_{i,j} > T$ in the optimal schedule.
  - Let

$$S_T \coloneqq \{ (i,j) \ : \ i \in M, j \in J , \quad p_{i,j} \leq T \}$$

denote the pairs between which the assignments are allowed w.r.t. the guess T.

- Let t^* denote the optimal makespan and $T \in \mathbb{Z}^+$ be a guess for t^*.
  - Let $S_T := \{ (i,j) : i \in M, j \in J, p_{i,j} \leq T \}$

    denote the pairs between which the assignments are allowed w.r.t. the guess T.

- Then we have the modified feasibility LP defined ***for each possible*** T.
  - Any integral solution is contained as a feasible solution in one of these LPs.
  - For any $T \geq t^*$, LP-(T) is guaranteed to be feasible.

    Conversely, whenever LP-(T) is infeasible, then $T < t^*$ must hold.

$$\sum_{i:(i,j)\in S_T} x_{i,j} = 1, \qquad \forall j \in J, \qquad \text{LP}-(T)$$

$$\sum_{j:(i,j)\in S_T} p_{i,j} \cdot x_{i,j} \leq T, \qquad \forall i \in M,$$

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

- Then we have the modified feasibility LP defined ***for each possible*** T.
  - Any integral solution is contained as a feasible solution in one of these LPs.
  - For any $T \geq t^*$, LP-(T) is guaranteed to be feasible.

    Conversely, whenever LP-(T) is infeasible, then $T < t^*$ must hold.
  - Under the ***parametric search framework***, it suffices to show that,

    whenever LP-(T) is feasible,
    we can always round the solution properly.

$$\sum_{i:(i,j)\in S_T} x_{i,j} = 1, \qquad \forall\, j \in J, \qquad \text{LP}-(T)$$

$$\sum_{j:(i,j)\in S_T} p_{i,j} \cdot x_{i,j} \leq T, \qquad \forall i \in M,$$

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

Parametric Search for Machine Scheduling

- We will derive a rounding process for LP-(T) such that the resulting makespan is at most $2T$.
  - Then, we can apply binary search to find the smallest T for which LP-(T) is feasible, and it follows that $T \leq t^*$.

    Then applying the rounding process gives us a 2-approximation.

$$\sum_{i:(i,j)\in S_T} x_{i,j} = 1, \qquad \forall\, j \in J, \qquad \text{LP}-(T)$$

$$\sum_{j:(i,j)\in S_T} p_{i,j} \cdot x_{i,j} \leq T, \qquad \forall i \in M,$$

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

The Extreme Point Structure for LP-(T)

Extreme Point Solutions for LP-(T)

$$\sum_{i:(i,j)\in S_T} x_{i,j} = 1, \qquad \forall\, j \in J, \qquad \text{LP}-(T)$$

$$\sum_{j:(i,j)\in S_T} p_{i,j} \cdot x_{i,j} \le T, \qquad \forall i \in M,$$

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

- The intuition here is that, although LP-(T) may have a number of variables, it has only a linear number of nontrivial constraints.
  - i.e., it has only $|J| + |M|$ constraints bounding the variables in a nontrivial way.
  - Hence, only a linear number of non-trivial variables can be defined at the extreme points of this LP.

    In other words, most of the variables must be set zero there.
  - Let $n = |J|$ and $m = |M|$.

Lemma 3.

Any extreme point solution to LP-(T) has at most $n + m$ nonzero variables.

$$\sum_{i:(i,j)\in S_T} x_{i,j} = 1, \qquad \forall\, j \in J, \qquad \text{LP}-(T)$$

$$\sum_{j:(i,j)\in S_T} p_{i,j} \cdot x_{i,j} \le T, \qquad \forall i \in M,$$

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

- Lemma 3 is a formal statement of the intuitions in the previous slide.
- The proof is straightforward.
  - Consider any extreme point solution of LP-(T) and the invertible matrix obtained from LP-(T) at that point.
  - At most $|J| + |M| = n + m$ nontrivial constraints can be selected to form the invertible matrix.
    Hence, the remaining constraints are from $x_{i,j} \ge 0$ and will set the corresponding variables to zero.

$$\sum_{i:(i,j)\in S_T} x_{i,j} = 1, \qquad \forall\, j \in J, \qquad \text{LP}-(T)$$

$$\sum_{j:(i,j)\in S_T} p_{i,j} \cdot x_{i,j} \le T, \qquad \forall i \in M,$$

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

- The following is a direct corollary of Lemma 3.

Corollary 4.

Any extreme point solution to LP-(T) must assign at least $n - m$ jobs integrally.

- Intuitively, Corollary 4 says that, at most m jobs are fractionally assigned.
  - The integrally-assigned jobs have a makespan of at most T.
  - Each of the fractionally-assigned jobs will contribute a makespan of at most T. (Since $x_{i,j} > 0$ implies that $p_{i,j} \le T$.)

  We will show that, ***there exists a matching*** from the fractionally-assigned jobs to the machines, and hence, those jobs can be properly assigned.

- The following is a direct corollary of Lemma 3.

Corollary 4.

Any extreme point solution to LP-(T) must assign at least $n - m$ jobs integrally.

- The proof of Corollary 4 is also simple.

  See the lecture note for the details.

$$\sum_{i:(i,j)\in S_T} x_{i,j} = 1, \qquad \forall\, j \in J, \qquad \text{LP}-(T)$$

$$\sum_{j:(i,j)\in S_T} p_{i,j} \cdot x_{i,j} \le T, \qquad \forall i \in M,$$

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

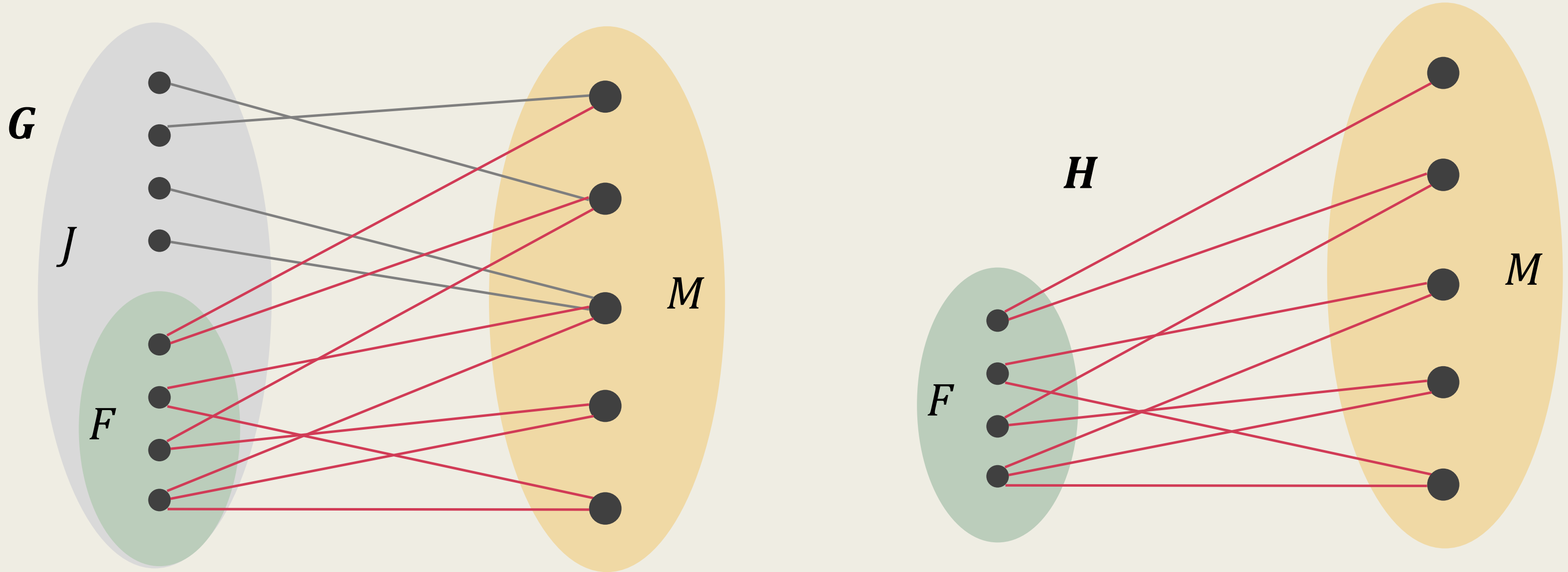
The Assignment Graphs and Properties

The Assignment Graph G and H

- Let $\boldsymbol{x}$ be an extreme point solution for LP-(T).

  Define the bipartite graph $G = (J, M, E)$ with partite set J and M such that $(j, i) \in E$ if and only if $x_{i,j} \neq 0$.

  Let $F \subseteq J$ be the set of jobs that are fractionally assigned in $\boldsymbol{x}$, and H be the subgraph of G induced by $F \cup M$.

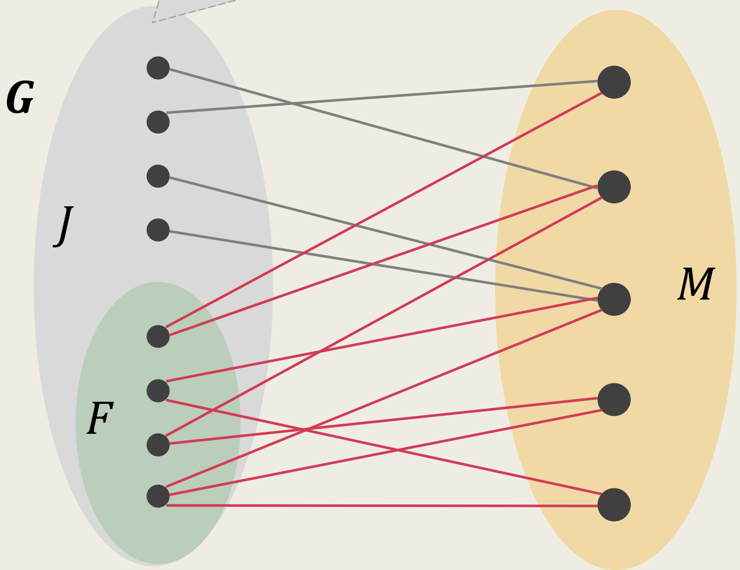


- Let $\boldsymbol{x}$ be an extreme point solution for LP-(T).

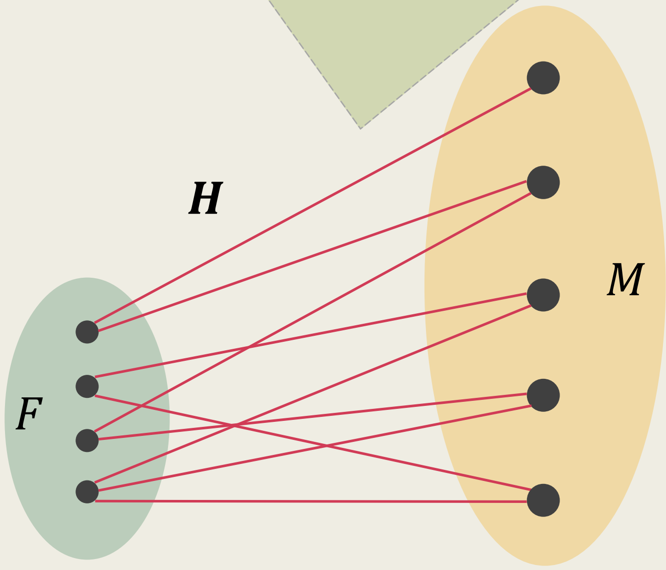
  Define the bipartite graph $G = (J, M, E)$ with partite set J and M such that $(j, i) \in E$ if and only if $x_{i,j} \neq 0$.

  Let $F \subseteq J$ be the set of jobs that are fractionally assigned in $\boldsymbol{x}$, and H be the subgraph of G induced by $F \cup M$.

Jobs in $J - \mathrm{F}$ have degree 1 in G and contribute a total makespan of $\leq T$.



Each edge (j, i) in H satisfies $p_{i,j} \leq T$.
Provided that there exists a matching to M exists, they will contribute a total makespan of $\leq T$.



We will show that, there is a **matching** from jobs in $\boldsymbol{F}$ **to** $\boldsymbol{M}$.

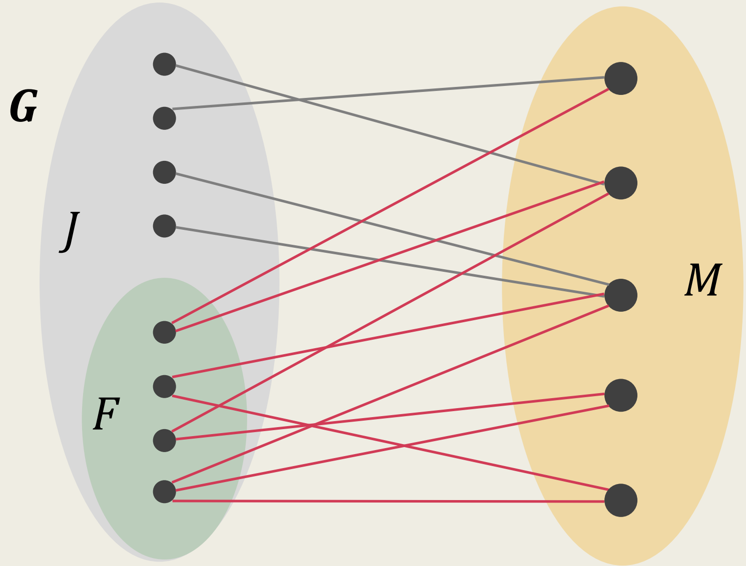
Then, we get a schedule with makespan $\leq 2T$.

G and H are Pseudo-Forests.

- We say that a connected graph with vertex set V is a pseudo-tree if it has at most $|V|$ edges.
  - i.e., it is either a tree, or a tree plus one edge.
- We say that a graph is a pseudo-forest if each of its connected components is a pseudo-tree.

Lemma 5.

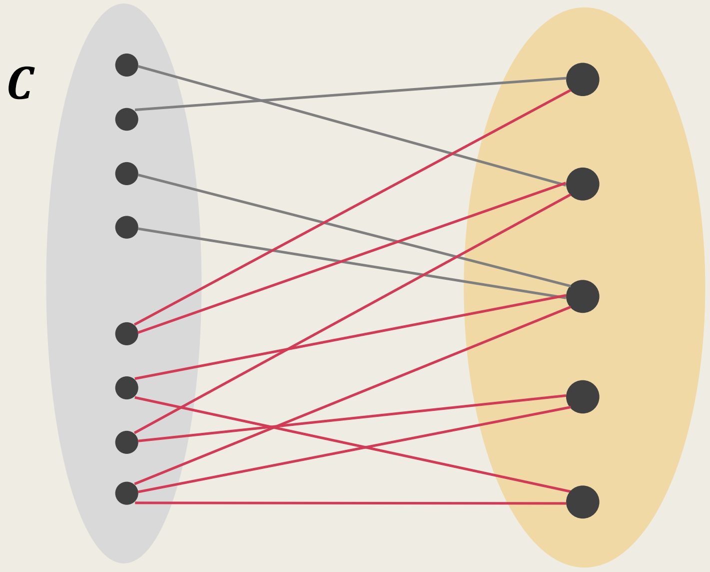
G is a pseudo-forest.



- Consider each connected component in G.
  - We will argue that it's a pseudo-tree.

Lemma 5.

G is a pseudo-forest.



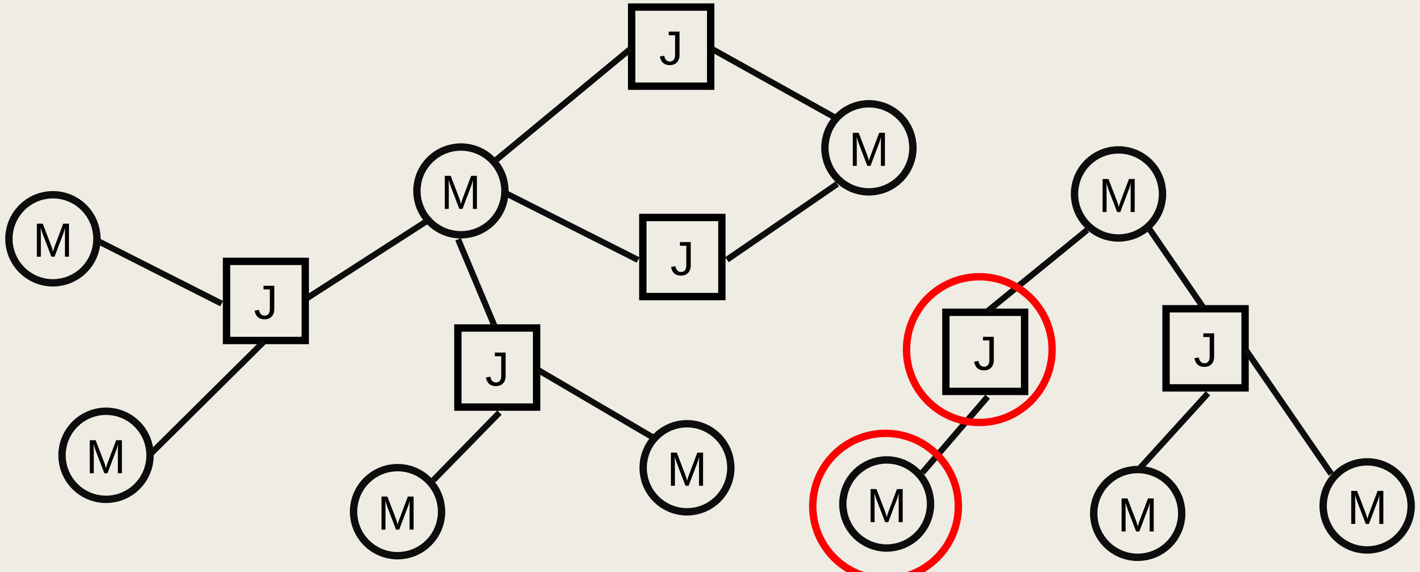
- Consider any connected component, say, C, in G.
  - Consider the variables and constraints to which C corresponds.

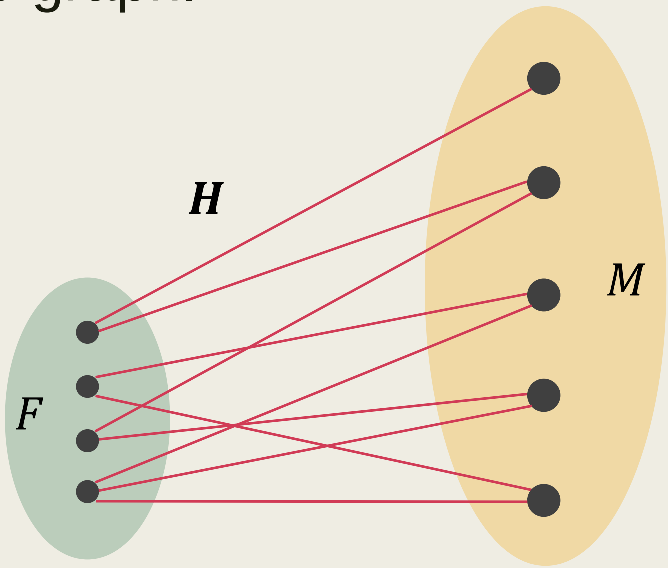
    Denote the sub-LP by LP-$(T)_C$.

    Clearly,
    the solution $\boldsymbol{x}$ restricted to C, $\boldsymbol{x}|_C$, must also be extreme for LP-$(T)_C$.
  - Hence, C has an equal number of vertices and edges and is a pseudo-tree.
- Since H is obtained by removing some degree-1 vertices from G, it is also a pseudo-forest.

H has a perfect matching (for F to M).

- We have shown that H is a pseudo-forest.
  - Since each job vertex in H has degree at least 2, we know that all the leaf vertices in H are machine vertices.
- The idea is to ***keep matching <u>a leaf machine vertex</u> with its parent <u>job vertex</u>***, and then remove both from the graph.



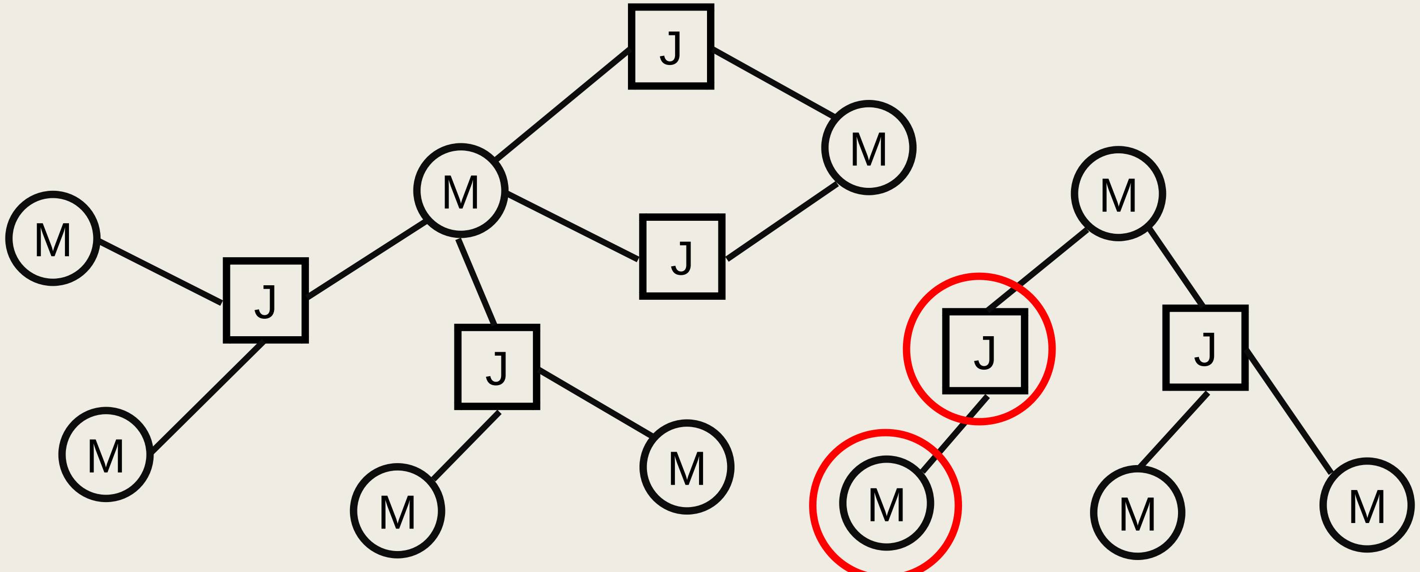


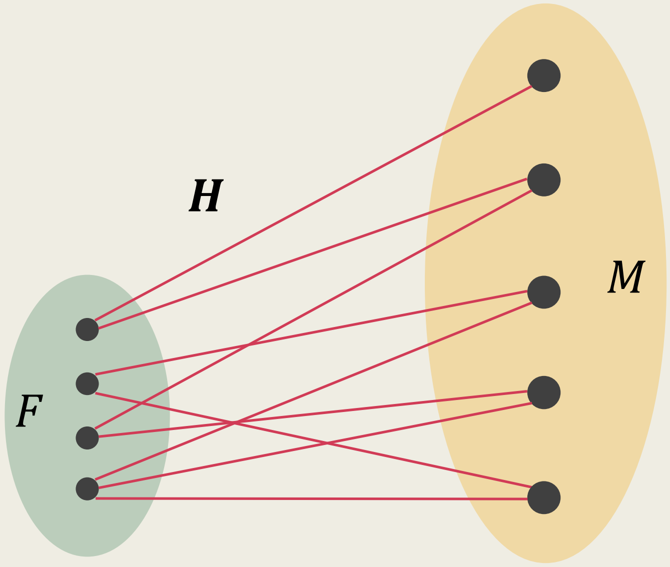
- Since each job vertex in H has degree at least 2, we know that all the leaf vertices in H are machine vertices.

- We repeat the following process until no more leaf vertex is left.
  - Pick a leaf machine vertex and match it with its parent job vertex.
  - Remove both vertices from the graph.

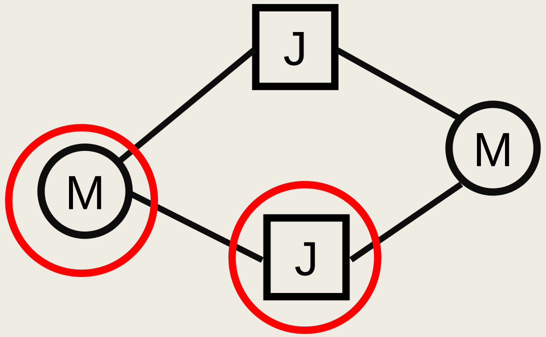
    Remove isolated vertices.

- Since this process does not change the degree of any other job vertex, the remaining leaf vertices are still machine vertices, H is still pseudo-forest.



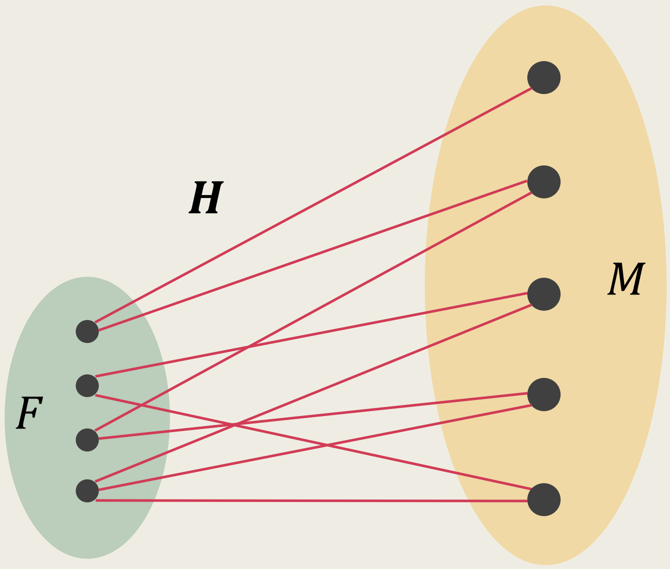


- We repeat the following process until no more leaf vertex is left.
  - Pick a leaf machine vertex and match it with its parent job vertex.
  - Remove both vertices from the graph.
    Remove isolated vertices.
- When this process ends,
  H is left with even cycles, which can be perfectly matched.



Lemma 6.

H has a perfect matching for F to M.



The Rounding Algorithm $\mathcal{A}$

Rounding the Extreme Point Solutions for LP-(T)

- The rounding algorithm $\mathcal{A}$ goes as follows.
  - Input: a basic feasible (extreme point) solution $\boldsymbol{x}$ for LP-(T)
  - Output: a schedule with makespan at most $2T$

1. Assign all the jobs in $J - F$ according to x.

   This contributes a makespan of $\leq T$.

2. Construct the graph H and compute a perfect matching from F to M. Assign the jobs in F according to the matching M.

   This also contributes a makespan of $\leq T$, since each machine gets at most one job with $p_{i,j} \leq T$.

3. Output the resulting schedule.

The 2-approximation algorithm for Unrelated Machine Scheduling

The 2-Approximation Algorithm

- The algorithm goes as follows.

This guarantees that $T \leq t^*$.

1. Apply binary search on $\left[0, \sum_{i,j} p_{i,j}\right]$ to find the smallest T such that LP-(T) is feasible.

2. Compute an extreme point solution $\boldsymbol{x}$ for LP-(T).

3. Apply rounding algorithm $\mathcal{A}$ on $\boldsymbol{x}$ and output the resulting schedule.

The output has a makespan of at most $2T \leq 2t^*$.