UNIVERSITY OF TWENTE.

Scheduling with Integer Programming and Project

Matthias Walter

Topics:

- Original-space models
- Extended formulations
- Time discretization

Preknowledge:

- Polyhedra

Agenda

1. Single Machine Scheduling
   - Single Machine Scheduling Problem
   - Original Space Description
   - Extended Formulation
   - Time Discretization
2. Project (Last Assignment)
   - Description

Agenda

1. Single Machine Scheduling
   - Single Machine Scheduling Problem
   - Original Space Description
   - Extended Formulation
   - Time Discretization

2. Project (Last Assignment)
   - Description

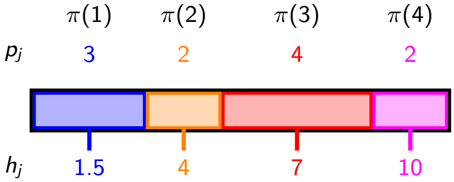
Single Machine Scheduling Problem

Definition – Permutation, half times

Consider a set J of jobs J with processing times $p_j > 0$ for each $j \in J$. A schedule of all jobs on a single machine without idle times is determined by its **permutation** $\pi : \{1, 2, \dots, n\} \to J$ which assigns a position in the schedule to the job. For a permutation π, the **half time vector** $h^\pi \in \mathbb{R}^J$ specifies when half of each job is carried out. It is the average of the **starting time vector** and the **completion time vector**.

Proposition – Half times for a permutation

For a permutation π, the half times are given by $h^\pi_{\pi(k)} = \sum_{\ell < k} p_{\pi(\ell)} + \frac{1}{2} p_{\pi(k)}$.



Smith's Rule

Theorem – Smith's Rule **[Smith '56]**

Let $w \in \mathbb{R}^J$. A schedule minimizes $w(h^\pi) := \sum_{j \in J} w_j h_j^\pi$ if and only if

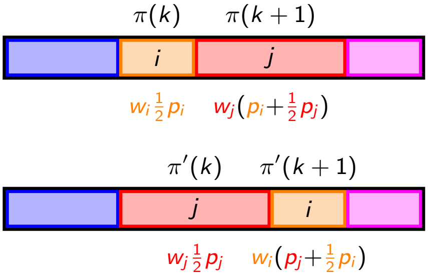
$$\frac{w_i}{p_i} \geq \frac{w_j}{p_j} \text{ holds for all } i, j \in J \text{ with } h_i^\pi < h_j^\pi. \tag{1}$$

Proof:

- Suppose (1) is not satisfied for some permutation π. Then there must exist jobs i and j that violate (1) and are scheduled directly after another.
- Let π' be equal to π, except that $\pi(k) = i$, $\pi(k+1) = j$ but $\pi'(k) = j$ and $\pi'(k+1) = i$ hold for a suitable position k.
- The objective value change is

$$w(h^{\pi'}) - w(h^\pi) = w_i(h_i^{\pi'} - h_i^\pi) + w_j(h_j^{\pi'} - h_j^\pi)$$
$$= w_i\left(p_j + \frac{p_i}{2} - \frac{p_i}{2}\right) + w_j\left(\frac{p_j}{2} - p_i - \frac{p_j}{2}\right) = w_i p_j - w_j p_i < 0.$$

Example:



- This contradicts optimality of π.
- Reverse direction: swapping jobs with "=" in (1) does not change the objective value. ■

Agenda

1. Single Machine Scheduling
   - Single Machine Scheduling Problem
   - Original Space Description
   - Extended Formulation
   - Time Discretization
2. Project (Last Assignment)
   - Description

Half Time Polytope

Definition – Half time polytope

The **half time polytope** is the convex hull of all half time vectors

$$P_{\text{ht}} := \text{conv}\left\{ h^{\pi} \in \mathbb{R}_{+}^{J} : \pi \text{ permutation} \right\}.$$

Some constraints:

$$\sum_{j \in S} p_j x_j \geq g(S) \text{ for all } \varnothing \neq S \subsetneq J \tag{2a}$$

$$\sum_{j \in J} p_j x_j = g(J) \tag{2b}$$

- where $g(S) := \min \left\{ \sum_{j \in S} p_j x'_j : x' \in P_{\text{ht}} \right\}$ for $S \subseteq J$.

The Value of $g(S)$

Corollary – Consequence of Smith's Rule

We have $g(S) := \min\left\{ \sum_{j \in S} p_j x'_j : x' \in P_{\text{ht}} \right\} = \frac{1}{2} p(S)^2$ for every $S \subseteq J$.

Proof:

- We minimize with weights $w_j := p_j$ for $j \in S$ and $w_j := 0$ for $j \notin S$.
- By Smith' Rule, we have to sort the jobs by w_j / p_j, which is either 1 ($j \in S$) or 0 ($j \notin S$).
- The objective value of any such permutation is

$$
\begin{aligned}
& w_{s_1} \tfrac{1}{2} p_{s_1} + w_{s_2}(p_{s_1} + \tfrac{1}{2} p_{s_2}) + \cdots + w_{s_k}(p_{s_1} + \cdots + p_{s_{k-1}} + \tfrac{1}{2} p_{s_k}) + 0 \\
&= p_{s_1} \tfrac{1}{2} p_{s_1} + p_{s_2}(p_{s_1} + \tfrac{1}{2} p_{s_2}) + \cdots + p_{s_k}(p_{s_1} + \cdots + p_{s_{k-1}} + \tfrac{1}{2} p_{s_k}) \\
&= \tfrac{1}{2}(p_{s_1} + p_{s_2} + \cdots + p_{s_k})^2 = \tfrac{1}{2} p(S)^2. \quad \blacksquare
\end{aligned}
$$

Constraints:

(2a): $\sum_{j \in S} p_j x_j \geq g(S)$

$\forall \emptyset \neq S \subsetneq J$

(2b): $\sum_{j \in J} p_j x_j = g(J)$

Corollary – Validity of inequalities

Every half time vector h^π satisfies (2).

Proof:

- The $\geq$-inequalities are satisfied by definition.
- The minimum in $g(J)$ is attained by all h^π, so (2b) holds. $\blacksquare$

Supermodularity of g

Lemma – supermodularity

$g = \frac{1}{2}p(S)^2$ is supermodular, i.e., $g(S \cap T) + g(S \cup T) \geq g(S) + g(T) \quad \forall S, T \subseteq J$.

Proof:

$$\begin{aligned} g(S \cap T) + g(S \cup T) &= \tfrac{1}{2}p(S \cap T)^2 + \tfrac{1}{2}p(S \cup T)^2 = \tfrac{1}{2}p(S \cap T)^2 + \tfrac{1}{2}\big(p(S \setminus T) + p(T \setminus S) + p(S \cap T)\big)^2 \\ &= \tfrac{2}{2}p(S \cap T)^2 + \tfrac{1}{2}p(S \setminus T)^2 + \tfrac{1}{2}p(T \setminus S)^2 + p(S \setminus T) \cdot p(T \setminus S) \\ &+ p(S \setminus T) \cdot p(S \cap T) + p(T \setminus S) \cdot p(S \cap T) \\ &= \tfrac{1}{2}p(S)^2 + \tfrac{1}{2}p(T)^2 + p(S \setminus T) \cdot p(T \setminus S) \geq g(S) + g(T). \quad \blacksquare \end{aligned}$$

Lemma – Uncrossing

Let $x \in \mathbb{R}^J$ satisfy (2) for $S \cup T$ and for $S \cap T$, and assume it satisfies (2) for S and for T with equality. Then $S \subseteq T$ or $T \subseteq S$.

Proof:

- Coefficient-wise comparison, the assumptions and the proof above imply:

$$0 = \sum_{j \in S \cap T} p_j x_j + \sum_{j \in S \cup T} p_j x_j - \sum_{j \in S} p_j x_j - \sum_{j \in T} p_j x_j \geq g(S \cap T) + g(S \cup T) - g(S) - g(T) = p(S \setminus T)p(T \setminus S).$$

Perfect Formulation for the Half Time Polytope

Theorem – Half time polytope **[Queyranne '91]**

The half time polytope is the set of vectors $x \in \mathbb{R}^J$ that satisfy (2).

Proof:

- Let $x \in \mathbb{R}^J$ be a vertex of (2).
- Hence, there exist (at least) n tight constraints (2) corresponding to sets $S_1, S_2, \dots, S_n \subseteq J$.
- By the uncrossing lemma, these sets form a chain, i.e., $S_1 \subsetneq S_2 \subsetneq \cdots \subsetneq S_n$.
- Since $S_1 \neq \emptyset$, we have $S_k = \{j_1, j_2, \dots, j_k\}$ for all $k \in \{1, 2, \dots, n\}$, where $J = \{j_1, j_2, \dots, j_n\}$.
- Tightness of the inequalities yields $p_{j_1} x_{j_1} = g(\{j_1\}) = \frac{1}{2} p_{j_1}^2$ and thus $x_{j_1} = \frac{1}{2} p_{j_1}$.
- For S_2 we have $p_{j_2} x_{j_2} = g(\{j_1, j_2\}) - g(\{j_1\})$ follows.
- By induction, also $x_{j_k} = \sum_{i<k} p_{j_i} \frac{1}{2} p_{j_k}$ follows for $k \in \{2, 3, \dots, n\}$.
- We conclude that x is a half time vector. ■

Separation Problem

Reminder: $\sum_{j \in S} p_j x_j \geq \min\{\sum_{j \in S} p_j x'_j : x' \in P_{\text{ht}}\}$ for all $S \subseteq J$ (2)

Theorem – Separation of inequalities **[Queyranne '91]**

The separation problem for (2) can be solved in $\mathcal{O}(n \log n)$ time.

Proof:

- A point $\hat{x} \in \mathbb{R}^J$ violates (2a) if and only if there exists a set $S \subseteq J$ with

$$\Gamma(S) := g(S) - \sum_{j \in S} p_j \hat{x}_j \quad > \quad 0.$$

- Add an element $k \notin S$ to S yields

$$\begin{aligned}
\Gamma(S \cup \{k\}) &= \tfrac{1}{2} p(S \cup \{k\})^2 - \sum_{j \in S \cup \{k\}} p_j \hat{x}_j \\
&= \tfrac{1}{2} p(S)^2 + p(S) p_k + \tfrac{1}{2} p_k^2 \quad - \quad \sum_{j \in S} p_j \hat{x}_j - p_k \hat{x}_k \\
&= \Gamma(S) + p_k (p(S) + \tfrac{1}{2} p_k - \hat{x}_k).
\end{aligned}$$

- Thus, if S is a maximizer and $k \notin S$, this implies $p(S) + \frac{1}{2} p_k - \hat{x}_k \leq 0$, which is equivalent to $\hat{x}_k \geq p(S) + \frac{1}{2} p_k$.

Separation Problem

Theorem – Separation of inequalities **[Queyranne '91]**

The separation problem for (2) can be solved in $\mathcal{O}(n \log n)$ time.

Proof (continued):

- Similarly, removing an element $k \in S$ yields

$$
\begin{aligned}
\Gamma(S \setminus \{k\}) &= \tfrac{1}{2} p(S \setminus \{k\})^2 - \sum_{j \in S \setminus \{k\}} p_j \hat{x}_j \\
&= \tfrac{1}{2} p(S)^2 - \tfrac{1}{2} p_k^2 - p(S \setminus \{k\}) p_k \quad - \quad \sum_{j \in S} p_j \hat{x}_j + p_k \hat{x}_k \\
&= \Gamma(S) + p_k(-\tfrac{1}{2} p_k - p(S \setminus \{k\}) + \hat{x}_k)
\end{aligned}
$$

- Thus, if S is a maximizer and $k \in S$, this implies $-\frac{1}{2} p_k - p(S \setminus \{k\}) + \hat{x}_k \leq 0$, which is equivalent to $\hat{x}_k \leq p(S) - \frac{1}{2} p_k$.
- Hence, S is a maximizer if and only if $\hat{x}_k \leq p(S) \iff k \in S$ holds for all $k \in J$.
- This shows that $\hat{x}_j \leq \hat{x}_k$ and $k \in S$ imply $j \in S$.
- Assuming that the jobs are $J = \{1, 2, \ldots, n\}$ with $\hat{x}_1 \leq \hat{x}_2 \leq \cdots \leq \hat{x}_n$, we only have to test violation for sets $S_k := \{1, 2, \ldots, k\}$ for $k = 1, 2, \ldots, n - 1$. ∎

Agenda

Extended Formulation

Variables:

- $x_j \in \mathbb{R}$ for $j \in J$ denotes the half time for job j.
- $y_{i,j} \in \{0,1\}$ for $i,j \in J$, $i \neq j$, indicates whether job i is scheduled before job j ($y_{i,j} = 1$) or not ($y_{i,j} = 0$).

Constraints:

$$\tfrac{1}{2} p_j + \sum_{i \in J \setminus \{j\}} p_i y_{i,j} = x_j \quad \forall j \in J \tag{3a}$$

$$y_{i,j} + y_{j,i} = 1 \quad \forall i,j \in J,\ i \neq j \tag{3b}$$

$$y_{i,j} \geq 0 \quad \forall i,j \in J,\ i \neq j \tag{3c}$$

Lemma – Isomorphism to cube

The feasible region of (3) is affinely isomorphic to a cube of dimension $\binom{n}{2}$.
In particular, all vertices have binary y-entries.

Proof:

- Eliminate the x-variables since they are just determined by (3a).
- Eliminate all $y_{i,j}$-variables for $i > j$ since they are determined by (3b).
- What remains is $0 \leq y_{i,j} \leq 1$ for all $i,j \in J$ with $i < j$. ∎

Extended Formulation

Theorem – Extended formulation for half time polytope **[Wolsey '80s]**

The half time polytope has an extended formulation of size $\mathcal{O}(n^2)$.

Proof:

- Let Q denote the polytope defined by (3).
- Each half time vector has a preimage by setting $y_{i,j} = 1$ if and only if $h_i < h_j$.
- It remains to show that every vertex (x, y) of Q (which has binary y-entries due to the lemma) satisfies $x \in P_{\text{ht}}$.
- Let $D = (J, A)$ be the digraph on node set J with $(i, j) \in A \iff y_{i,j} = 1$.
- If D contains no directed cycle, then it corresponds to a schedule, and x is its half time vector.
- Assume that D contains a directed cycle. Since for every $\{i, j\}$, either $(i, j) \in A$ or $(j, i) \in A$ holds, D must even contain a directed cycle of length 3.

Extended Formulation

Reminder: $x_j = \frac{1}{2} p_j + \sum_{i \in J \setminus \{j\}} p_i y_{i,j}$ (3a)

Proof (continued):

- Consider the cycle $C = \{(i,j),(j,k),(k,i)\} \subseteq A$.
- If we replace the arc (i,j) by (j,i), we obtain the new vector $x^{i,j} \in \mathbb{R}^J$ with

$$x^{i,j}_\ell - x_\ell = \begin{cases} p_j & \text{if } \ell = i, \\ -p_i & \text{if } \ell = j, \\ 0 & \text{otherwise,} \end{cases}$$

  and similar formulas hold for $x^{j,k}$ (replacing (j,k) by (k,j)) and $x^{k,i}$ (replacing (k,i) by (i,k)).
- We obtain

$$\big(p_k(x^{i,j}_\ell - x_\ell) + p_i(x^{j,k}_\ell - x_\ell) + p_j(x^{k,i}_\ell - x_\ell)\big) \;\; / \;\; \big(p_i + p_j + p_k\big) = 0 \qquad \ell = 1,2,\dots,n$$

  and thus

$$\frac{p_k}{p_i + p_j + p_k} x^{i,j}_\ell + \frac{p_i}{p_i + p_j + p_k} x^{j,k}_\ell + \frac{p_j}{p_i + p_j + p_k} x^{k,i}_\ell = x,$$

  which shows that x is a convex combination of other feasible solutions, and thus not a vertex of the projection. ■

Agenda

1. Single Machine Scheduling
   - Single Machine Scheduling Problem
   - Original Space Description
   - Extended Formulation
   - Time Discretization

2. Project (Last Assignment)
   - Description

Time Discretization

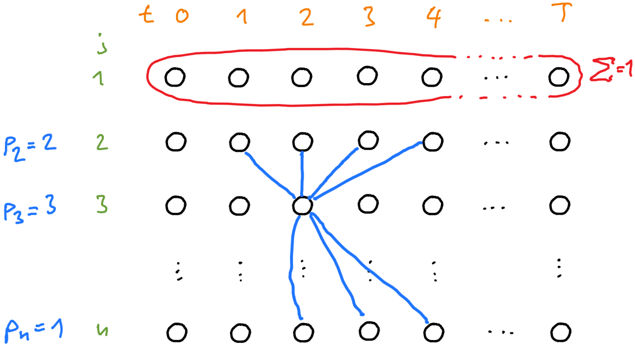
Idea:

- To avoid continuous variables, we try to model the problem with only binary variables.
- We **discretize** the allowed starting times, e.g., $s_j \in \mathcal{T} := \{0, 1, 2, \ldots, T\}$.
- In general this is an approximation, but not here due to $p_j \in \mathbb{Z}$.

Variables:

- $y_{j,t} \in \{0,1\}$ for $(j,t) \in J \times \mathcal{T}$: $y_{j,t} = 1 \iff$ job j starts **at** time t.

Visualization:



Time Discretization Formulation

Variables:

- $y_{j,t} \in \{0,1\}$ for $(j,t) \in J \times \mathcal{T}$: $y_{j,t} = 1 \iff$ job j starts **at** time t.

IP:

$$\min \sum_{j \in J} \sum_{t \in \mathcal{T}} w_j (t + \tfrac{1}{2} p_j) y_{j,t} \tag{4a}$$

$$\text{s.t.} \quad \sum_{t \in \mathcal{T}} y_{j,t} = 1 \qquad \forall j \in J \tag{4b}$$

$$y_{j,t} + y_{j',t'} \leq 1 \qquad \forall (j,t),(j',t') \in J \times \mathcal{T} : j \neq j', t' \in (t - p_{j'}, t + p_j) \tag{4c}$$

$$y_{j,t} \in \{0,1\} \qquad \forall (j,t) \in J \times \mathcal{T} \tag{4d}$$

Proposition – Correctness of time-discretized formulation

Formulation (4) correctly models the minimum weight half time scheduling problem.

Proof:

- (4a) is correct: $y_{j,t} = 1$ means that j starts at t and thus is half done at $t + \frac{1}{2} p_j$.
- (4b) ensures that each job is started exactly once.
- (4c) ensures that jobs j and j' do not overlap. ∎

Agenda

Multiple Sub-Jobs on Different Matchines

Assignment 5 – Project

We consider a **job shop scheduling problem** with the following input data:

- Number m of machines $\texttt{machines} = \{1, 2, \dots, m\}$.
- List $\texttt{jobs}$ of jobs.
- Processing times $\texttt{processingTimes}[k, j] \in \mathbb{Z}$ indicating the processing time of job j on machine k.

The goal is to schedule each job $j \in \texttt{jobs}$ on every machine $m \in \texttt{machines}$ such that

- each machine can only run one job at a time,
- each started job runs until it is completed,
- a job on machine $m \geq 1$ can only start once it is completed on machine $m - 1$.
- among all such schedules, the overall makespan (i.e., time until all jobs on machine m are completed) is minimized.

The task of this exercise is to create **two** implementations of MIP models that tackle this problem. These can be of completely different types, but they can also be a base model as well as the same model augmented with problem-specific cutting planes. Also price-and-branch methods (i.e., only price for the root LP) are allowed despite them being heuristic.

Lesson Recap – **Any Questions?**

1. Single Machine Scheduling
   - Single Machine Scheduling Problem
   - Original Space Description
   - Extended Formulation
   - Time Discretization
2. Project (Last Assignment)
   - Description