Scheduling with Storage Resources

Ch. Schwindt,
University of Karlsruhe (TH)

Outline
1. Problem
2. Model
3. Solution methods
   3.1 Branch-and-bound
   3.2 Priority-rule method
4. Conclusions
1 Problem

- Given:
  - Set of operations executed on dedicated processing units
  - Set of input and output products depleted and replenished in batch mode
  - Input and output quantities for operations
  - Storage facilities of finite capacity for stocking products
  - Initial stocks and prescribed safety stocks for products
  - Minimum and maximum time lags between start times of operations

- Sought:
  - Feasible production schedule minimizing some regular objective function
2 Model

- Notations
  - $O$: Set of operations $i = 0, 1, \ldots, n, n+1$ with processing times $p_i$ ($p_0 = p_{n+1} = 0$)
  - $\mathcal{R}$: Set of storage resources $k$ with safety stocks $R_k$ and storage capacities $\overline{R}_k$
  - $r_{ik}$: Increase in inventory level of resource $k$ by execution of operation $i$
  - $O_k^- = \{i \in O \mid r_{ik} < 0\}, O_k^+ = \{i \in O \mid r_{ik} > 0\}$: Sets of depleting and replenishing operations for resource $k$
  - $\delta_{ij}$: Time lag between linked operations $(i, j) \in E \subseteq O \times O$, distances $d_{ij}$
  - $S = (S_0, S_1, \ldots, S_{n+1})$: Production schedule
  - $r_k(S, t) = \sum_{i \in O_k^- : S_i \leq t} r_{ik} + \sum_{i \in O_k^+ : S_i + p_i \leq t} r_{ik}$: Inventory in resource $k$ at time $t$
  - $f : \mathbb{R}^{n+2}_{\geq 0} \rightarrow \mathbb{R}$: Regular objective function in start times $S_i$ ($i \in O$)

- Model
  
  \[
  \begin{align*}
  \text{Minimize} & \quad f(S) \\
  \text{subject to} & \quad R_k \leq r_k(S, t) \leq \overline{R}_k \quad (k \in \mathcal{R}, \ t \geq 0) \\
  & \quad S_j - S_i \geq \delta_{ij} \quad ((i, j) \in E) \\
  & \quad S_i \geq 0 \quad (i \in O)
  \end{align*}
  \]
### 3 Solution Methods

#### 3.1 Branch-and-Bound

- Scheduling is . . .
  - defining precedence relationships between operations competing for scarce resources (*Sequencing: hard*)
  - optimizing objective function subject to prescribed time lags and established precedence relationships (*Temporal scheduling: tractable*)

- Enumeration scheme
Resolving resource conflicts

- **Inventory excess** at time $t$: $r_k(S, t) > \bar{R}_k$

  ![Diagram of inventory excess](image)

  » Delay completion of some operation $j \in O^+_k$ up to start of some operation $i \in O^-_k$
  
  » Precedence relationship $S_j + p_j \geq S_i$: $\delta_{ij} = -p_j$ (maximum time lag)

- **Inventory shortage** at time $t$: $r_k(S, t) < \underline{R}_k$

  » Delay start of some operation $j \in O^-_k$ up to completion of some operation $i \in O^+_k$
  
  » Precedence relationship $S_j \geq S_i + p_i$: $\delta_{ij} = p_i$ (minimum time lag)
### 3.2 Priority-rule method

**Why classical priority-rule methods don’t work**

- Stepwise expand partial schedule by scheduling one eligible operation in each iteration
- Machine scheduling or project scheduling with **renewable resources**: partial schedules are feasible
- Scheduling with **storage resources**:
  - Material-availability constraints and storage-capacity constraints
  - Feasibility of partial schedules would require simultaneous scheduling of several operations

▷ Allow for **infeasible partial schedules**
Two-phase approach

- **Phase 1:** Scheduling subject to material-availability constraints
  - Relax storage-capacity constraints
  - Operation $j$ eligible if
    - all operations $i \in O$ with $d_{ij} \geq 0$ and $d_{ji} < 0$ have been scheduled
    - for resulting partial schedule, terminal inventories do not fall below safety stocks
  - Select some eligible activity $j^*$ according to priority indices $\pi(j)$
  - Schedule $j^*$ at time $t^* = \min\{t \geq ES_{j^*} \mid r_k(S^C, \tau) + r_{jk} \geq R_k \text{ for } k \in \mathcal{R}, \tau \geq t\}$

- **Pegging by precedence relationships according to FIFO strategy**
  - Iterate operations $i \in O_k^+$ in order of nondecreasing $S_i + p_i$ and allot output of $i \in O_k^+$ to operations $j \in O_k^-$ in order of nondecreasing $S_j$
  - Introduce time lag $\delta_{ij} = p_i$ between $i \in O_k^+$ and $j \in O_k^-$ consuming output of $i$

- **Phase 2:** Scheduling subject to precedence and storage-capacity constraints
  - Relax material-availability constraints
  - Proceed analogously to phase 1, replacing material-availability with storage-capacity constraints
### Serial schedule-generation scheme (phase 1)

\[
\begin{align*}
    u & := 0; \\
    2: \quad S_0 & := 0, \ C := \{0\}; \\
    \text{for all } i \in O & \text{ (*initialize } ES_i \text{ and } LS_i *) \\
    ES_i & := d_{0i}, \ LS_i := -d_{i0}; \\
    \text{while } \mathcal{C} \neq O & \text{ do} \\
    \quad \mathcal{E} & := \{j \in O \setminus \mathcal{C} \mid \text{Pred}(j) \subseteq \mathcal{C}, \ \sum_{i \in \mathcal{C} \cup \{j\}} r_{ik} \geq R_k \text{ for } k \in \mathcal{R}\}; \\
    \quad \text{if } \mathcal{E} = \emptyset & \text{ then terminate;} \\
    \quad j^* & := \min\{j \in \mathcal{E} \mid \pi(j) = \text{ext}_{h \in \mathcal{E}} \pi(h)\}; \\
    \quad t^* & := \min\{t \geq ES_{j^*} \mid r_k(S^c, \tau) + r_{j^*, k} \geq R_k \text{ for } k \in \mathcal{R}, \ \tau \geq t\}; \\
    \quad \text{if } t^* > LS_{j^*} & \text{ then (*unschedule and restart*)} \\
    \quad \quad u & := u + 1; \\
    \quad \quad \text{if } u > \bar{u} & \text{ then terminate;} \\
    \quad \quad \mathcal{U} & := \{i \in \mathcal{C} \mid LS_{j^*} = S_i - d_{j^*i}\}; \\
    \quad \quad \text{for all } i \in \mathcal{U} & \text{ do } d_{0i} := S_i + t^* - LS_{j^*}; \\
    \quad \quad \text{update distances } d_{ij} & \text{ for all } i, j \in O \text{ and goto line 2;} \\
    \text{else (*schedule } j^* \text{ at time } t^* *) \\
    \quad S_{j^*} & := t^*, \ \mathcal{C} := \mathcal{C} \cup \{j^*\}; \\
    \quad \text{for all } j \in O \setminus \mathcal{C} & \text{ do (*update } ES_j \text{ and } LS_j *) \\
    \quad \quad ES_j & := \max(ES_j, S_{j^*} + d_{j^*j}); \\
    \quad \quad LS_j & := \min(LS_j, S_{j^*} - d_{jj^*}); \\
    \text{return } S;
\end{align*}
\]
<table>
<thead>
<tr>
<th>No.</th>
<th>FWS</th>
<th>FWS_1</th>
<th>GB</th>
<th>Term. S</th>
<th>GB Term. S</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>*</td>
<td>32</td>
<td>24</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>32</td>
<td>*</td>
<td>32</td>
<td>36</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>32</td>
<td>*</td>
<td>32</td>
<td>36</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>24</td>
<td>*</td>
<td>24</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>44</td>
<td>*</td>
<td>44</td>
<td>72</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>44</td>
<td>*</td>
<td>44</td>
<td>72</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>44</td>
<td>*</td>
<td>44</td>
<td>72</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>16</td>
<td>24</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>16</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>16</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>16</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>12</td>
<td>24</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>12</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>12</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>12</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>8</td>
<td>24</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>8</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>8</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>8</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>6</td>
<td>24</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>6</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>6</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>6</td>
<td>48</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

**WFS**

Storage capacity and storage time settings FIS, PWG.

- 36 problem instances with 12 to 90 operations
- Randomized multi-pass priority rule based method
- Branch-and-bound algorithm

**4 Experimental performance analysis**

Scheduling with Storage Resources
5 Conclusions

- **Scheduling with storage resources**
  - Operations consuming input products and producing output products
  - Prescribed safety stocks and limited storage capacities for products
  - Minimum and maximum time lags between operations

- **Branch-and-bound method**
  - Relax inventory constraints
  - Branch over alternative precedence relationships resolving resource conflicts

- **Priority-rule method**
  - Two-phase method
  - Ensure material availability by precedence relationships

- **Talk by Norbert Trautmann**
  - Renewable resources
  - Sequence-dependent changeover times on processing units
  - Performance analysis comparing priority-rule to branch-and-bound method