Scheduling Theory and Applications

Prof. B.M.T. Lin (林妙聰)
Department of Information and Finance Management
Institute of Information Management
National Chiao Tung University, TAIWAN
May 2008
Outline

- Introduction and Overview
- Success Stories
- Production Scheduling
  - Single Machine
  - Parallel Machine
  - Shops and Flexible Assembly Systems
  - Supply-Chain Scheduling, Scheduling on the Internet and Green Logistics
- Workforce Planning and Timetabling
- Linking Real-world Applications with Academic Research
- Research Methods

Source:
The Role of Scheduling

- Allocation of limited resources to tasks over time
- A decision-making process that has as a goal of the optimization of one or more objectives
Success Story:
The US Army National Guard Simulators Location and Routing Problem (Murty and Djang, 1999)
Decisions

- **Home Base Selection**: Select home bases for the mobile trainers from among the suitable sites.
- **Secondary Training Site Selection**: Select the necessary secondary training sites so that each armory is within MTD of at least one of the home bases or secondary training sites.
- **Routing of Mobile Trainers**
- **Scheduling of Training Sessions**
Objectives

- Mobile Trainer Fleet Mileage
- Platoon-Bus Mileage
- Number of Secondary Training Sites
Solution Models

- $p$-Median Problem for Home Base Selection
- Set-Cover Problem for Secondary Training Site Selection
- Vehicle Routing Problem for Routing of Mobile Trainers and Scheduling of Training Sessions
Results

- Reduction of mobile trainer mileages from 321,000 miles to 72,850 miles
- Cost saving mobile trainer mileage is $123M
- A one-time cost saving accrues from a reduction in the number of armory facility upgrades is $3M
Success Story:
SLIM: Short Cycle Time and Low Inventory in Manufacturing at Samsung Electronics
(Leachman, Kang and Lin, 2002)
The Problem

- SEC is a victim of its own success due to long the cycle time the fabrication process
- The original paradigm for scheduling production lots: urgency or due dates of lots to be scheduled
- Solution: Switch the paradigm to focusing on the capacity of each individual device
Results and Economic Benefits

- Cycle time required by fabricating 64MB RAM was reduced from 80 or more days to 27 days.
- Total sales revenue of DRAM was $21.9 billions, among which $954 million was gained through the reduction of cycle time.
- Total benefit 1 billion.
- Market share increased from 18 ~ 22 percents.
- President Y. W. Lee: *SLIM is essential to the success of SEC in the semiconductor business.*
More Applications

- College/University Class(room) Scheduling
- NBC’s Ads Scheduling
- Crew Recovery and Pilot Training at Continental Airlines
- Delta optimizes continuing-qualification-training schedules for pilots
- Scheduling Umpire Crews for Professional Tennis Tournaments
- Scheduling the Beef Fabrication Process at Swift & Company
- Scheduling of Refuel at NY Nuclear Power Plant
- Sequence-dependent Scheduling at Baxter International
Framework (Deterministic Models)

- Framework: three-field notation $\alpha|\beta|\gamma$
- $\alpha$: machine configuration
- $\beta$: job characteristics and/or constraints
- $\alpha$: objective function
Framework (Deterministic Models)

- **Jobs**
  - **Static data**:
    - $p_{ij}$: processing time of job $j$ on machine $i$.
    - $r_{ij}$: release date of job $j$ on machine $i$
    - $d_j$: due-date/deadline of job $j$
    - $w_j$: weight (importance indicator) of job $j$
  - **Dynamic data**:
    - starting time ($S_{ij}$)
    - completion time ($C_{ij}$)
Framework (Deterministic Models)

Machine Configurations

- Single machine (1)
- Identical machines in parallel ($Pm$)
- Machines in parallel with different speeds ($Qm$)
- Unrelated Machines in parallel ($Rm$)
- Flow shop ($Fm$)
- Flexible flow shop ($FFs$)
- Open shop ($Om$)
- Job shop ($Jm$)
- Supply chain models
Framework (Deterministic Models)
Processing Characteristics and Constraints

- Release dates
- Sequence-dependent setup times/costs
- Preemptions
- Precedence constraints
- Breakdown and maintenance constraints
- Machine eligibility constraints
- Permutation
- Blocking/No-wait
Framework (Deterministic Models)
Processing Characteristics and Constraints

- Recirculation/Reentrant
- Workforce constraints
- Routing constraints
- Material handling constraints
- Make-to-stock and make-to-order
- Transportations
Framework (Deterministic Models)

Performance Measures and Objectives

- Throughput and makespan objectives $C_{\text{max}}$
- Total (weighted) completion time: $\sum w_j C_j$, $\sum C_j$
- Due-date related objectives
  - Maximum lateness, $\max \{C_j - d_j\}$
  - Total (weighted) tardiness, $\sum w_j \max \{0, C_j - d_j\}$
  - Total late work, $\sum \max \{p_j, \max \{0, C_j - d_j\}\}$
  - (Weight) Number of tardy jobs, $\sum w_j U_i \sum U_i$
  - Earliness-Tardiness $\sum \alpha_j E_j + \sum \beta_j T_j$
- Setup costs
- Work-in-process (WIP) inventory costs
- Finished goods inventory costs
- Transportation costs
Example Problems

- $1 \mid r_i \mid \sum U_i$
- $1 \mid \text{prec} \mid \sum w_i C_i$
- $Pm \mid \text{tree} \mid L_{\text{max}}$
- $Fm \mid \mid C_{\text{max}}$
- $Om \mid r_{ij} \mid C_{\text{max}}$
Service Models

- Activities are concerned: meeting, flight leg, game, appointment, personnel position
- Typically no goods to inventorize
- The amount of resources may vary over time
Operational Characteristics and Constraints

- Time windows (release dates and due dates)
- Capacity requirements and constraints
- Preemptions
- Operator and tooling requirements
- Workforce scheduling constraints
Performance Measures and Objectives

- Makespan
- Setup costs
- Earliness and tardiness costs
- Convenience costs and penalty
- Personnel costs
Single Machine

- Only a single server is available
- At any time, only one job/activity can be processed
- Consider the following example:
  - Each job has a processing time $p_i$ and a release date $r_i$.
  - We want to find a feasible schedule such that the sum of completion times is minimized.

The problem is NP-hard, but solvable in polynomial time when preemption is allowed.
Parallel Machines

- A job/activity can be processed by any of the machines
- Machines may have different processing speeds/capabilities
- Most of the standard objectives are NP-hard
Flow Shops

- Flow shop is a set of \((m)\) machines that are arranged in a pipeline fashion.
- Each job \(J_i\) has \(m\) operations in which operation \(O_{ij}\) must be processed on machine \(M_j\) with processing time \(p_{ij}\).
- Processing of all jobs must follow the route \(M_1, M_2, \ldots, M_m\).
Job Shops

- A set of \( \{m\} \) machines are available
- Each job \( J_i \) has \( m \) or less operations
- Different jobs may have different processing routes
- e.g. job \( \leftarrow \rightarrow \) patient
SCM Scheduling
Medium-term Planning + Short-term Scheduling
Inventory cost, transportation cost, tardiness/shortage cost, … are involved

Scheduling on the Internet

Green Logistics
Interval Scheduling, Reservation, Timetabling and Workforce Scheduling
Off-day scheduling

Crew scheduling
Sports Scheduling, Train Timetabling
Linking Real-world Applications with Academic Research
A Model for Fire Engine Manufacturing

Machine $M_1$

Machine $M_2$

Machine $M_3$

Machine $M_4$

Machine $M_5$
## Redevelopment Project in Boston

<table>
<thead>
<tr>
<th>$H$</th>
<th>$h_1$</th>
<th>$h_2$</th>
<th>$h_3$</th>
<th>$h_4$</th>
<th>$h_5$</th>
<th>$h_6$</th>
<th>$h_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>21</td>
<td>7</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>$b_i$</td>
<td>7</td>
<td>5</td>
<td>15</td>
<td>17</td>
<td>16</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>$\delta_i$</td>
<td>3</td>
<td>-3</td>
<td>10</td>
<td>-4</td>
<td>9</td>
<td>-7</td>
<td>-7</td>
</tr>
</tbody>
</table>

Sequence $S = h_1 h_3 h_5 h_4 h_6 h_7 h_2$

<table>
<thead>
<tr>
<th>$t$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_t$</td>
<td>7</td>
<td>10</td>
<td>20</td>
<td>29</td>
<td>25</td>
<td>18</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>$\delta_i$</td>
<td>$h_1$</td>
<td>$h_3$</td>
<td>$h_5$</td>
<td>$h_4$</td>
<td>$h_6$</td>
<td>$h_7$</td>
<td>$h_2$</td>
<td></td>
</tr>
</tbody>
</table>
### Redevelopment Project in Boston

#### Table:

<table>
<thead>
<tr>
<th>$H$</th>
<th>$h_1$</th>
<th>$h_2$</th>
<th>$h_3$</th>
<th>$h_4$</th>
<th>$h_5$</th>
<th>$h_6$</th>
<th>$h_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_i$</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>21</td>
<td>7</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>$a_i$</td>
<td>7</td>
<td>5</td>
<td>15</td>
<td>17</td>
<td>16</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>$\delta_i$</td>
<td>3</td>
<td>-3</td>
<td>10</td>
<td>-4</td>
<td>9</td>
<td>-7</td>
<td>-7</td>
</tr>
</tbody>
</table>

**Sequence** $S = h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7$

<table>
<thead>
<tr>
<th>$t$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_t$</td>
<td>7</td>
<td>10</td>
<td>7</td>
<td>17</td>
<td>X</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\sigma t}$</td>
<td>$h_1$</td>
<td>$h_2$</td>
<td>$h_3$</td>
<td>$h_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Film Production Problem

Total cost = 2*810+3*700+4*910 + 880+5*400+3*610

<table>
<thead>
<tr>
<th>actor &lt;i&gt;&lt;i&gt;</th>
<th>Shooting day &lt;i&gt;&lt;i&gt;</th>
<th>Holding Cost &lt;c&gt;&lt;i&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11100100</td>
<td>810</td>
</tr>
<tr>
<td>2</td>
<td>11000110</td>
<td>700</td>
</tr>
<tr>
<td>3</td>
<td>01010001</td>
<td>910</td>
</tr>
<tr>
<td>4</td>
<td>01011110</td>
<td>880</td>
</tr>
<tr>
<td>5</td>
<td>01000001</td>
<td>400</td>
</tr>
<tr>
<td>6</td>
<td>00110001</td>
<td>610</td>
</tr>
</tbody>
</table>
### Film Production Problem

<table>
<thead>
<tr>
<th>actor $i$</th>
<th>Shooting day $j$</th>
<th>Holding Cost $c_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1 1 0 0 1 0 1</td>
<td>810</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 0 0 1 1 1</td>
<td>700</td>
</tr>
<tr>
<td>3</td>
<td>1 1 0 1 0 0 0 0</td>
<td>910</td>
</tr>
<tr>
<td>4</td>
<td>0 1 0 1 1 1 1 0</td>
<td>880</td>
</tr>
<tr>
<td>5</td>
<td>1 1 0 0 0 0 0 0</td>
<td>400</td>
</tr>
<tr>
<td>6</td>
<td>1 0 1 1 0 0 0 0</td>
<td>610</td>
</tr>
</tbody>
</table>

Total cost = $3\times810 + 3\times700 + 1\times910 + 880 + 610$
Runway Scheduling at Rome Airport
Research Methods

• Modeling and Formulation
• **Settling time complexity** (NP-hardness Vs. polynomial solvability)
• **Exact solution methods**
  • Branch-and-bound algorithms (optimality properties, bounding functions, dominance rules)
  • Dynamic programming algorithms
• **Approximation Approach**
  • Heuristics: computational & analytical performances
  • Meta-heuristics/Local Search: computational study
  • Approximability/non-approximability
  • Approximation scheme
Q & A
Proof of NP-hardness

**Theorem 1.** The RPD\(^+\)_2 problem is NP-hard in the strong sense.

**Proof.** To establish the NP-hardness of the two-machine relocation problem, we present a polynomial-time reduction from the NMTS problem. Suppose that we are given two sets of integers \(\{x_1, x_2, \ldots, x_n\}\), and \(\{y_1, y_2, \ldots, y_n\}\), and a collection of targets \(B_1, B_2, \ldots, B_n\). Without loss of generality, we assume \(B_1 \leq B_2 \leq \cdots \leq B_n\). We group all targets with the same value together in a subset. Assume there are \(k\) different target values and let \(m_i\) be the cardinality of \(i\)th subset. We get \(B_1 = \cdots = B_{m_1} < B_{m_1+1} = \cdots = B_{m_1+m_2} < \cdots < B_{m_1+m_2+\cdots+m_{k-1}+1} = \cdots = B_{m_1+m_2+\cdots+m_k} = B_n\). Let \(M = 1 + \max_{i=1,...,k} m_i\). Set \(\tilde{B}_i = B_{m_1+m_2+\cdots+m_i}\) for all \(1 \leq i \leq k\).

We define an instance \(I\) of RPD\(^+\)_2 as follows. In instance \(I\), there are two machines, \(2n\) basic jobs \(J_{1i}, i = 1, 2, \ldots, n\), and \(J_{2i}, i = 1, 2, \ldots, n\), and \(k - 1\) connecting jobs \(\tilde{J}_i, i = 1, \ldots, k - 1\). All connecting jobs and jobs \(J_{1i}, i = 1, 2, \ldots, n\), have to be executed on machine one, and jobs \(J_{2i}, i = 1, 2, \ldots, n\), have to be executed on machine two. For each integer \(x_i\), we create basic job \(J_{1i}\) with \(\alpha_{1i} = 2Mx_i\) and \(\beta_{1i} = 2Mx_i + 1\). For each integer \(y_i\), we create basic job \(J_{2i}\) with \(\alpha_{2i} = 2My_i\) and \(\beta_{2i} = 2My_i + 1\). Note that

\[
\beta_{1i} > \alpha_{1i} > ME/2 \quad \text{for all } i = 1, \ldots, n.
\]  

(5)

We use \(\tilde{\alpha}_i, \tilde{\beta}_i, \tilde{\delta}_i\) to denote the amount of required resources, amount of returned resources and net contributions of connecting job \(\tilde{J}_i\). For each connecting job \(\tilde{J}_i\), define \(\tilde{\alpha}_i = 2M\tilde{B}_i + 2m_i\) and \(\tilde{\beta}_i = 2M\tilde{B}_{i+1}\). Note that \(\tilde{\delta}_i = 2M\tilde{B}_{i+1} - (2M\tilde{B}_i + 2m_i) = 2M(\tilde{B}_{i+1} - \tilde{B}_i) - 2m_i \geq 2M - 2m_i > 0\). The inequality implies \(\tilde{\alpha}_i < \tilde{\alpha}_{i+1}\) for all \(i = 1, \ldots, k - 2\). Moreover, we have

\[
\tilde{\alpha}_i = 2M\tilde{B}_i + 2m_i > 2M\tilde{B}_1 = 2MB_1 > ME.
\]  

(6)
<table>
<thead>
<tr>
<th>n</th>
<th>Opt_zero</th>
<th>LBA</th>
<th>Opt_zero</th>
<th>LBT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>Avg</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0.01</td>
<td>0.04</td>
<td>0.07</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>0.91</td>
<td>8.06</td>
<td>28.72</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>17.23</td>
<td>417.66</td>
<td>697.82</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1,290.98</td>
<td>1,290.98</td>
<td>1,290.98</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>45</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>65</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Algorithm CB-TCT:
Initial conditions:
$h(n + 1, u) = 0$ for all $u, 1 \leq u \leq n$.

Recursion:

$$h(i, u) = \min_{i \leq j \leq n} \left\{ h(j + 1, u + 1) + (j - i + 1) \times \max_{1 \leq k \leq m} \left\{ (u + 1)s_k + \sum_{l=1}^{j} p_{kl} \right\} \right\}$$

Goal:
Find $h(1, 0)$. 
Approximation Algorithm and Non-approximability

Splitting Algorithm.

1. Let $I$ be an instance of RP. Apply Rounding Algorithm to $I$ and let $\sigma$ be the derived schedule.
2. Let $m$ be the number of machines and $n_t$ be the number of jobs scheduled in interval $[t, t + 1)$ for $t = 0, \ldots, Z(\sigma) - 1$.
   
   For all $t$ such that $n_t > m$ do:
   
   Set $\tau := \lceil \frac{t + 1}{m} \rceil$.
   
   Schedule $n_t$ jobs in intervals $[t, t + 1), [t + 1, t + 2), \ldots, [t + \tau - 1, t + \tau)$ such that each of the first $\tau - 1$ intervals contain $m$ jobs.
   
   For each job $J_i$ with $C_i(\sigma) \geq t + 1$ set $C_i(\sigma) = C_i(\sigma) + \tau$.
3. Stop.

**Theorem 5.** Splitting Algorithm is a $(3 - \frac{2}{m})$-approximation algorithm for the RP problem with $m$ identical parallel machines.

**Proof.** Let $\sigma$ be the schedule obtained by Splitting Algorithm. We say that time interval $[t, t + 1)$ is complete if exactly $m$ jobs are allocated within this interval in schedule $\sigma$, otherwise time interval $[t, t + 1]$ is incomplete. Let $h$ and $l$ be the numbers of complete and incomplete intervals correspondingly, i.e. $Z(\sigma) = l + h$.

Note that after the splitting of “overloaded” unit time intervals, we get at most one new unit time interval that has less than $m$ jobs. It follows that $l$ does not exceed the length of schedule obtained by Rounding Algorithm. From Theorem 4 we have that $l \leq 2 \times \text{OPT}$. On the other hand, for any instance of the relocation problem with $n$ UET jobs and $m$ machines we get $\frac{hm + l}{m} \leq \text{OPT}$. The inequality follows from the fact that each complete interval contains exactly $m$ jobs and each incomplete interval contains at least one job. Combining the above facts, we thus get

\[
\frac{Z(\sigma)}{\text{OPT}} \leq \frac{l + h}{\max\{\frac{l}{2}, \frac{hm + l}{m}\}}.
\]
Approximation Algorithm and Non-approximability

Theorem 3. Unless P = NP, there is no ρ-asymptotic approximation algorithm for the \( \text{RP}_\infty \) problem for any \( \rho < \frac{4}{3} \), even if \( \alpha_i \leq \beta_i \) and \( p_i = 1 \) for all \( i \).

Proof. We consider an instance \( I \) of the Partition problem: Given a finite set of integers \( X = \{e_1, e_2, \ldots, e_n\} \) with \( \sum_{e_i \in X} e_i = 2E \), is there a subset \( X_1 \subseteq X \) such that \( \sum_{e_i \in X_1} e_i = E \)?

We define an instance \( I' \) of the \( \text{RP}_\infty \) problem as follows. Let \( K > 0 \) be a fixed integer. We introduce \( n(K + 1) \) basic jobs \( J_{ki} \) with \( \alpha_{ki} = (E + 1)^k e_i \) and \( \beta_{ki} = (E + 1)^k e_i + \frac{1}{n} \) for all \( k = 0, 1, \ldots, K \) and \( i = 1, 2, \ldots, n \). Note that for convenience parameter \( \beta_{ki} \) does not abide by the integer constraint. The violation can be resolved by scaling all parameters with a factor \( n \). The jobs are grouped into \( K + 1 \) sets of \( n \) jobs with the same first index, i.e., set \( N_k = \{J_{k1}, J_{k2}, \ldots, J_{kn}\} \). We define \( K \) connecting jobs \( \tilde{J}_k, 1 \leq k \leq K \), with \( \tilde{\alpha}_k = E(E + 1)^{k-1} + 1 \) and \( \tilde{\beta}_k = E(E + 1)^k \). The amount of initial resources is \( \Omega_0 = E \).

We use the notion of schedule concatenation as follows: \( \sigma = N_0N_1N_2 \ldots N_l \) means that the jobs of set \( N_i \) are followed by the jobs of set \( N_{i+1} \) in schedule \( \sigma \) for all \( i = 0, 1, 2, \ldots, l - 1 \). It is easy to see that in any feasible schedule \( \sigma \), the jobs are processed in the following order \( \sigma = N_0\{\tilde{J}_1\}N_1\{\tilde{J}_2\}N_2\{\tilde{J}_3\}N_3 \ldots \{\tilde{J}_K\}N_K \). Since the amount of initial resources is equal to \( E \), only the jobs from set \( N_0 \) are available for processing at time zero. Moreover, since the net contribution of each basic job is equal to \( \frac{1}{n} \), only the connecting job \( \tilde{J}_1 \) becomes available when all jobs from set \( N_0 \) finish their processing. After finishing job \( \tilde{J}_1 \), the amount of resources is equal to \( E(E + 1) \) and we can schedule only jobs from set \( N_1 \). This line of reasoning continues for the remaining jobs (Fig. 2). In schedule \( \sigma \), at the time job \( J_i, i = 1, \ldots, K \) is to be processed, it will consume all available resources. It follows that no job can be processed in parallel with a connecting job \( J_i \). Now consider the jobs from any set \( N_k, 0 \leq k \leq K \). They require \( \sum_{i=1}^n \alpha_{ki} = \sum_{i=1}^n (E + 1)^k e_i = 2E(E + 1)^k \) units of resources. The amount of resources available after finishing jobs in \( N_0\{J_1\}N_1\{J_2\}N_2\{J_3\}N_3 \ldots \{\tilde{J}_K\} \) is equal to \( E(E + 1)^k \). Since \( \delta_{ki} = \frac{1}{n} \) and all \( e_i \) are integers we can schedule the jobs of set \( N_k \) in time interval of length 2 if and only if there exists a partition of the job set...