IE 505 PRODUCTION PLANNING AND CONTROL

Hakesh Nagi

na@buffalo.edu
University at Buffalo - State University of New York
Department of Industrial Engineering

Lectures Notes

All other rights reserved
© For limited distribution to IE 505 students
Introduction

Operations Scheduling

Dynamic versus Static Scheduling

Project Scheduling

Vendor Scheduling

Vehicle Scheduling

Facilities Scheduling

Personal Scheduling

Types of Scheduling Problems:

- Sequencing: determining the order of processing jobs
- Assigning start (and/or) end times to activities on resources
- Determining a time-phased plan of activities on resources
- Allocating available capacity or resources to jobs through time

Job-Shop Scheduling: Flow Shop
Time-phasing of Jobs/Operations
Job-Shop Scheduling

Time-phased schedule for components
Materials Requirements Planning (MRP)

Schedule of products with time
Master Production Scheduling (MPS)

Work-Force Levels
Aggregate Production Planning

Demand Forecasting

Hierarchy of Production Planning Decisions
Operations Scheduling
5. Evaluation of alternatives
4. Material flow and sequencing
3. Number of workers in the shop
2. Number and types of machines
1. Job arrival (loading) pattern

A. Issues

Terminology •
Objectives •
Issues •

Characteristics of Job-Shop Scheduling
8. Minimize production and worker costs
7. Minimize set-up times
6. Reduce setup times
5. Provide accurate job-status
4. Maximize utilization
3. Minimize average flow time, wait time
2. Minimize work-in-process (WIP)
1. Meet due-dates: Min. Earliness, Tardiness, Lateness, # of Tardy Jobs

B. Objectives (often conflicting)

Characteristics of Job-Shop Scheduling
6. Tardiness, earliness and lateness
5. Make-span
4. Flow time
3. Parallel versus sequential processing
2. Job shop
1. Flow shop
C. Terminology

Characteristics of Job-Shop Scheduling
\[
\frac{\text{remaining time}}{\text{job processing time}} = CR
\]

1. FCFS (First-Come-First-Served)
2. SPT (Shortest Processing Time)
3. EDD (Earliest Due Date)
4. CR Critical Ratio

Simple Sequencing Rules
SEQUENCING THEORY FOR A SINGLE MACHINE
Number of permutations: $u$

$$\frac{2^u}{u!} = \mathcal{P}$$

Mean Flow time:

$$\{u\mathcal{L}, \cdots, u\mathcal{L}\} \mathcal{M}_{\text{max}} = x \mathcal{L}$$

Maximum Tardiness:

$$[0, u\mathcal{T}] \text{ max} = T_{\text{Earliness}} = u \mathcal{P}$$

$$[0, u\mathcal{T}] \text{ max} = T_{\text{Tardiness}} = u \mathcal{L}$$

$$u \mathcal{P} - u \mathcal{L} = \text{ Lateness} = u \mathcal{T}$$

$$u \mathcal{P} + u \mathcal{L} = \text{ Flow time} = u \mathcal{M}$$

Waiting time

Due date

Processing time
BASIC RESULTS: SINGLE MACHINE SCHEDULING
Which is clearly minimized by setting:

\[ t_1 \leq t_2 \leq \ldots \leq t_n \]

Mean Flow time:

\[ F' = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{t_k} \]

\[ = \frac{1}{n} \sum_{k=1}^{n} \left( \frac{1}{t_1} + \frac{1}{t_2} + \ldots + \frac{1}{t_n} \right) \]

\[ = \frac{1}{n} \left( \frac{1}{t_1} + \frac{1}{t_2} + \ldots + \frac{1}{t_n} \right) \]

Proof: Let \([1], [2], \ldots, [n]\) be a permutation:

Flow time of job at position \(k\) is:

\[ F[k] = \sum_{i=1}^{k} t_i \]

1. SPT minimizes Mean Flow time, \(F'\)
\[ T_{\text{max}}(\text{new}) \geq T_{\text{max}}(\text{old}) \]

If for some \( k \), we can show that by interchanging them,

\[ [u]p \geq \cdots \geq [z]p \geq [1]p \]

Proof: By interchange argument.

3. EDD minimizes maximum lateness, \( T_{\text{max}} \)

3. Mean Lateness

2. Mean Waiting time

1. Mean Flow time, \( F_1 \)

The following measures are equivalent.
Example

4. Optimal sequence = current sequence with rejected jobs appended.

4. Moore's Algorithm for Minimum # of Tardy Jobs

1. Sequence jobs in EDD order.

2. Find first tardy job, say [i]. If none, go to 4.

3. Consider jobs [i], [i+1], ... [i+j], and reject job with largest processing

3. Consider jobs [1], [2], ... [i], and reject job with largest processing

2. Return to step 2.

Due date 15 6 9 23 20 30

Job 1 3 4 5 6

10 3 4 8 10 6
Step 4. Optimal sequence is 2, 3, 4, 6, 1, 5.

<table>
<thead>
<tr>
<th>Job</th>
<th>Due date</th>
<th>Cj</th>
<th>Tj</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>9 23</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>9 23</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>23 30</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

Due date 6 9 23 30

Job 2

Job 5 has largest T*; Reject Job 5:

Job 4 is tardy; Consider 2, 3, 5 and 4

<table>
<thead>
<tr>
<th>Job</th>
<th>Due date</th>
<th>Cj</th>
<th>Tj</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>9 23</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>9 23</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>23 30</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>15 30</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

Due date 6 9 23 30

Job 2

Step 3. Job 3 has largest T*; Reject Job 3:

Job I is tardy; Consider 2, 3 and I

<table>
<thead>
<tr>
<th>Job</th>
<th>Due date</th>
<th>Cj</th>
<th>Tj</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>9 23</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>9 23</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>23 30</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>15 30</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>I</td>
<td>15 30</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

Due date 6 9 23 30

Job 2

Step 2. Job I is tardy; Consider 2, 3 and I

<table>
<thead>
<tr>
<th>Job</th>
<th>Due date</th>
<th>Cj</th>
<th>Tj</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>9 23</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>9 23</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>23 30</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>15 30</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>I</td>
<td>15 30</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

Due date 6 9 23 30

Job 2

Step 1. EDD rule

Basic Results: Single Machine Sequencing
\[ L = (\gamma p - \gamma H) = (\gamma H) \gamma \beta \]

\[ T = \gamma p - \gamma H = (\gamma H) \gamma \beta \]

Examples of \( \gamma \phi \) is non-decreasing in \( \gamma H \)

\[ \min_{\gamma H} \max_{\gamma p > \gamma n} \gamma \]

5. Precedence Constraints: Algorithm
Example: Minimize the Maximum Lateness

1. Draw the precedence constraint graph
2. Set $\mathcal{A} = \{\text{jobs with no successors}\}$
3. Set $T = \exists t | t \notin \mathcal{S}$
4. Schedule job $\forall g \in \mathcal{G}$ next from end
5. If more jobs left, go to 2, otherwise, stop.

Algorithm

Basic Results: Single Machine Sequencing

IE 505 Production Planning and Control
Job 5 (assignment 5)

Job 5 (assignment 6)

Precedence Graph
SEQUENCING ALGORITHMS FOR MULTIPLE MACHINES
Objective is to minimize the makespan or max flow time. Jobs must be processed in the order machine 1 then machine 2.

1. n jobs on 2 machine flow-shop

\[ \wedge (u) \leftarrow t \]

Tractable only in special cases

Increase of complexity: n jobs on m machines

Sequential Algorithms for Multiple Machines
4. Stop when done
3. Cross off scheduled job
   If it belongs to Col B, Schedule job last
   If it belongs to Col A, Schedule job next
2. Find smallest remaining value in the 2 columns
1. List values of \( A^i \) and \( B^i \) in 2 columns
\[
(Basic Rule: \text{Job } i \text{ precedes } j \text{ if } \text{min}(A^i, B^j) > \text{min}(A^j, B^i))
\]
\[
B^i = \text{processing time of job on} B
A^i = \text{processing time of job on} A
\]
Machine 1 = A; Machine 2 = B.

Same order on both machines.

Optimal schedule is a permutation schedule (un): Jobs must follow the

Johnson's Algorithm for \( n/2 \) in \( \text{max problem} \)

IE 305 Production Planning and Control
1. Job 4 scheduled: 4
2. Job 5 scheduled: 4
3. Job 2 scheduled: 4
4. Job 3 scheduled: 4

Example: \( \frac{H/2}{H/4_{\text{max}}} \)
\[ A' + B' + C' = \bar{B'} \quad \text{and} \quad \bar{B'} + A' + A = \bar{C'} \]

Then,

\[ \text{min}(A') \leq \text{max}(B') \]

3-machine problem can be reduced to 2-machine IF

Extension to 3 machine case n/3/\text{max problem}
4. Find total time, draw Gantt-Chart.
   • horizontal & vertical both jobs
   • vertical Job 2
   • horizontal Job 1

(possible (don't cross blocks))

3. Draw path from origin to upper right corner s.t. you go 45° as often as
   possible.

2. Block out rectangular areas where machine "used" by both jobs

1. Draw an X-Y graph: axes represent processing times of sequential
   operations of Jobs 1 and 2, respectively.

Solution not optimal.

2 jobs must be processed through an m-machine job-shop - "Smart

Aker's Graphical solution to 2/m/Cmax Problem
Example: $2/4|C_{\text{max}}|F_{\text{max}}$ Machines - A, B, C, D

<table>
<thead>
<tr>
<th>Job</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>C</td>
<td>D</td>
<td>C</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>C</td>
<td>B</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>B</td>
<td>1</td>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>A</td>
<td>6</td>
<td>A</td>
<td>4</td>
</tr>
</tbody>
</table>

Seq: Job 1, Job 2
STOCHASTIC SCHEDULING
Uses Principles of Queuing Theory

2. Dynamically arriving jobs

\[ [u]n/([u]t)H \geq \cdots \geq [z]n/([z]t)H \geq [l]n/([l]t)H \]

Single Machine Case: Minimized by Expected Weighted SPT:

\[
\left( \frac{1}{\pi} \sum_{u \rightarrow l} \left( \frac{1}{u} - \frac{1}{l} \right) H \right) H = H
\]

Criterion is an Expected Value, e.g. Mean Weighted Flow Time:

I. Stochastic processing times; Jobs known

Stochastic Scheduling
VEHICLE SCHEDULING
• AGV Routing Problem in FMS
• Dial-a-Ride Problem
• Distribution Model
• Chinese Postman Problem
• Traveling Salesman Problem

Practical Applications
Minimum number of workstations = $C_n(T/C) = \sum_{i=1}^{n} t_i$

Total Work Content:

- Objective: desired cycle time $C$
- Group tasks at stations: right balance
- $t_i$ is the time for each job
- $n$ jobs to be processed in an assembly line
4. Assume a "good" cycle-time $C$

3. Sort tasks in decreasing positional weight

$$f \in \text{success}(i) \quad f < i$$

2. Positional weight of $i = \sum_{f \in \text{success}(i)} f$

1. Determine the precedence network of tasks

Heuristic: Ranked positional weight technique

7. Stop, when satisfactory cycle-time and balance is obtained.

6. Try another $C$, i.e. go to 4.

5. Assign jobs from sorted list without violating capacity and precedence

4. Assume a "good" cycle-time $C$

3. Sort tasks in decreasing positional weight

2. Positional weight of $i = \sum_{f \in \text{success}(i)} f$

1. Determine the precedence network of tasks