Management Practice

11. Scheduling

Jeroen.Bergmann@eng.ox.ac.uk

MPiE

17/11/2020

Course

Literature for the course:

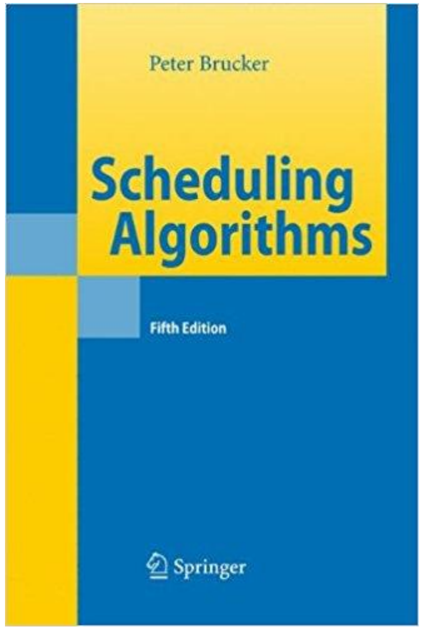
Eisner, Howard. *Essentials of project and systems engineering management*. John Wiley & Sons, 2008.

Learning objective for this session:

- Understand scheduling approaches
- Able to solve basic scheduling problems

Literature for this session:

www.springer.com/gb/book/9783540695158



Schedule management definition

- Schedule management is the process of developing, maintaining and communicating schedules for time and resource.
- A schedule is the timetable for a project, programme or portfolio. It shows how the work will progress over a period of time and takes into account factors such as limited resources and estimating uncertainty.

Source: The Chartered Body for the Project Profession (APM)

Schedule

- The scheduling process starts with the work that is needed **to deliver stakeholder requirements**. This includes the technical work that creates outputs, the change management work that delivers benefits, and the management activity that handles aspects, such as risk management and stakeholder management.
- Some types of work can be defined much more easily than other types. The work involved in building a house is clear from the start. The work involved in maintaining a generator is not clear until inspections are complete. Engineering work tends to have complete specifications from the start, whereas change management and some IT work follow a more iterative approach to defining what needs to be done.

Source: The Chartered Body for the Project Profession (APM)

Schedule

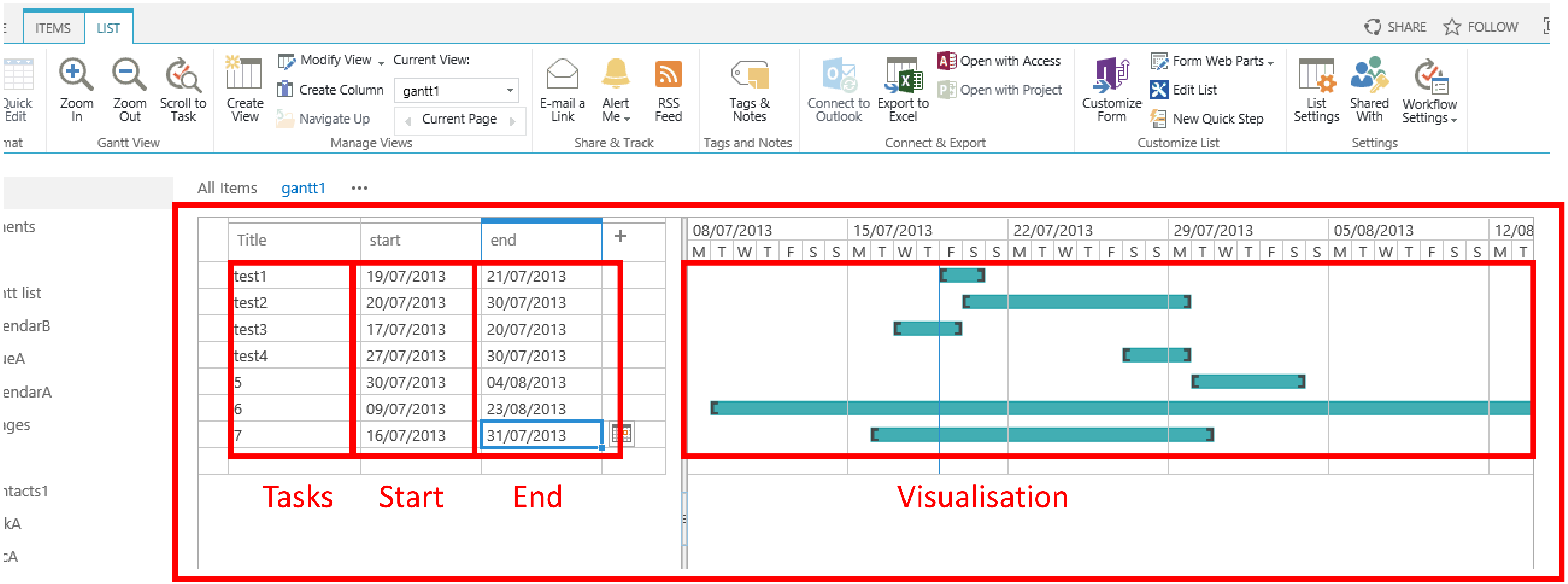
Schedules are presented in many different ways in order to suit the circumstances. The choice of presentation will depend upon:

- the level of detail required;
- whether time and/or resource is being shown;
- the context of the work (e.g. construction, IT, engineering or business change);
- the dimension being scheduled (project, programme or portfolio);
- the target audience.

The most common form of graphical schedule is the Gantt chart. In its simplest form this uses bars on a horizontal timescale to show the start, duration and finish of packages of work.

Source: The Chartered Body for the Project Profession (APM)

Gantt Chart - Recap



Person/group/machine

Source: Microsoft SharePoint

Example of job scheduling

- Processor scheduling
- Bandwidth scheduling
- Package delivery scheduling
- Patient admission scheduling

Finding the optimal sequence of jobs.

20 jobs on one machine gives 20! (2.43×10^{18})

Let’s look at scheduling again

- Scheduling is the process of organizing, choosing and timing resource usage to carry out all the activities necessary to produce the desired outputs at the desired times, while satisfying a large number of time and relationship constraints among the activities and the resources

Source: Morton and Pentico, 1993

Models in scheduling

- The assessment of ‘scheduling’ as defined by Critical Path Analysis (CPA) is now over 60 year old.
- The field to use algorithms really started in the fifties and then evolved further in the seventies, computer scientists started to use scheduling as a tool for improving the performance of computer systems.

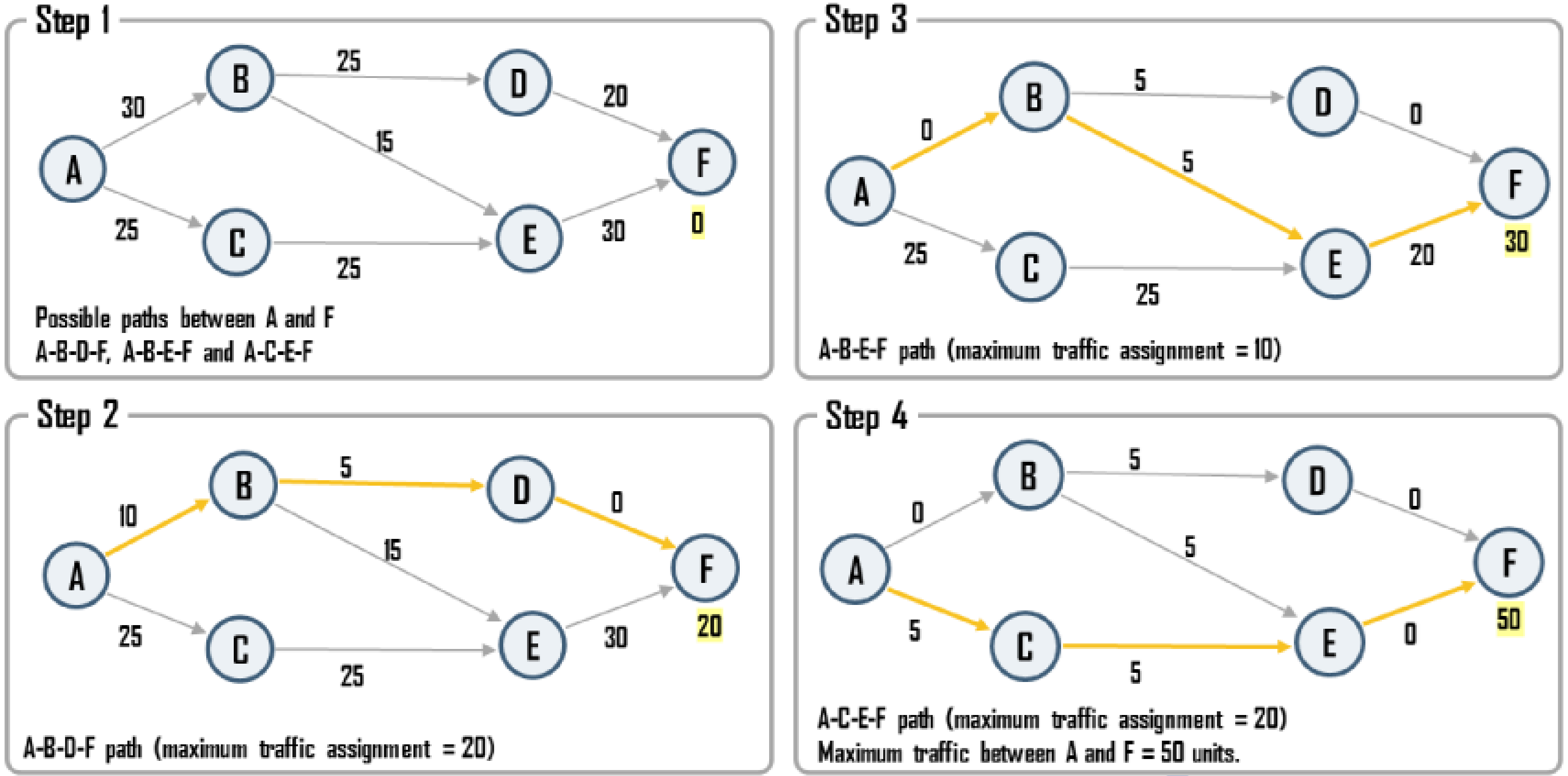
Source: Peter Brucker, Scheduling Algorithms, 2007

Models in scheduling

- **Instance:** Particular set of data for the model
- **Exact algorithm:** Optimum solution for every instance
- **Heuristic algorithm:** an acceptable solution, that is optimal or at least close to optimal for every instance

Heuristic example

- Assigning the maximum possible amount of traffic between two locations (A and F) can be solved **heuristically** (other options possible)

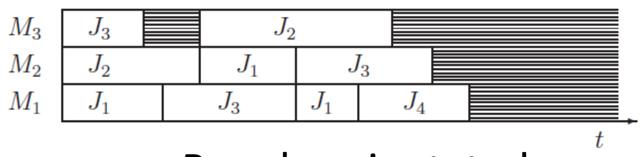


Source: Hofstra University, Geography of transport systems

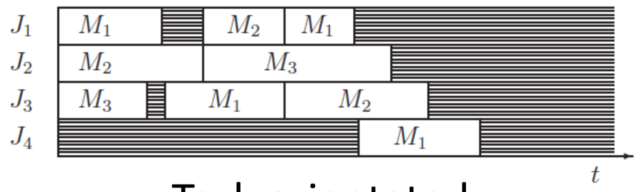
Models in scheduling

- Suppose we have *m* number of Machines or people M_j with $j=1,\ldots,m$ who need to process *n* Jobs or tasks J_i $(i=1,\ldots,n)$.

- This allows us to state; a **schedule** is for each job an allocation of one or more time intervals to one or more people.



People orientated



Task orientated

Source: Peter Brucker, Scheduling Algorithms, 2007

Job

- A job J_i consists of a number n_i of operations O_{ij} .
- Associated with operation O_{ij} is a **processing time** p_{ij} , specific for a person/machine (*i*) and job (*j*).

$$\textit{in the case } J_i \textit{ and } n_i = 1 \quad \therefore \quad J_i = O_{i1}$$

- If there is only one operation (n_i = 1), we then identify J_i with O_{i1} and denote the processing requirement by p_i. The job in this case it the single operation.

- A **release time** r_i, on which the first operation of J_i becomes available for processing may be specified.

Source: Peter Brucker, Scheduling Algorithms, 2007

Schedule is feasible

The schedule is **feasible** if:

(1) No two time intervals overlap on the same machine

(2) No two time intervals allocated to the same job overlap

(3) It meets a number of problem-specific characteristics.

A schedule is **optimal** if it minimizes a given optimality criterion.

Source: Peter Brucker, Scheduling Algorithms, 2007

Optimising the scheduling

- From a management point of view we want to **optimise cost and profit**, but it is hard to directly relate this to the schedule.
- In terms of project management it is essential that the project is **on time**, on budget and up to quality. Time is a leading factor in accomplishing this. The problem is how achieve it.
- A specific measurable outcome can thus be selected against which the schedule is optimised. One suitable candidate is time.
- A cost function $f_i(t)$ can be set which will be minimised for an certain optimality criteria. For example it can be completing J_i at time t.

Source: Peter Brucker, Scheduling Algorithms, 2007; Prince2

Scheduling problems

Classes of scheduling problems are specified in terms of a three-field classification:

- α specifies the **machine environment**
- β specifies the **job characteristics**
- γ denotes the **optimality criterion**

All three need to be specified to define the problem

Source: Graham et al. (1979)

Machine environment (α)

- Single Machine
- Parallel Machines (identical vs. different)
- Flow Shops: different machines (e.g. assembly lines)

– Each job must be processed by each machine exactly once

– All jobs have the same routing (same order)

– A job cannot begin processing on the second machine until it has completed the processing on the first

- Job Shops

– Each job may have its own routing (different routes can be selected)

- Open Shops (e.g. car repair shop)

– Jobs have no specific routing (but will need to be determined)

Machine environment (α)

The following parameters are set

$$\alpha 1 \in \{\circ, P, Q, R, PMPM, QMPM\}$$

If each job can be processed on each of the machines $M_1, \ldots, M_m$ then $\alpha 1 \in \{P, Q, R\}$

Scheduled jobs on machines have a certain flow-time and thus there is a speed (s).

P = identical parallel machines ($p_{ij} = p_i$)

Q = uniform parallel machines ($p_{ij} = \frac{p_i}{s_j}$) with s_j the speed of machine M_j

R = unrelated parallel machines ($p_{ij} = \frac{p_i}{s_{ij}}$) with job-dependent speeds s_{ij} of M_j

PMPM = multi-purpose machines with identical speeds

QMPM = multi-purpose machines with uniform speeds

Source: Peter Brucker, Scheduling Algorithms, 2007

Machine environment – shops

In a problem with **multi-purpose machines** a set of machines μ_j is associated with each job j indicating that j can be processed on <u>one</u> machine in μ_j only.

$$\alpha 1 \in \{G, J, F, O, X\}$$

This is a multi-operation model. Associated with each job J_i there is a set of operations. The machines are <u>dedicated</u>, i.e. all μ_{ij} are one element sets.

G = general shop, there are precedence relations between arbitrary operations.

There are m machines $M_1, \dots , M_m$ and n jobs $j = 1, \dots , n$.

Job j consists of $n(j)$ operations $O_{1j}, O_{2j}, \dots , O_{n(j)j}$ where O_{ij} must be processed for p_{ij} time units on a dedicated machine $\mu_{ij} \in \{M_1, \dots , M_m\}$.

Two operations of the same job **cannot** be processed at the same time. Precedence constraints are given between the operations.

Source: Peter Brucker, Scheduling Algorithms, 2007

Machine environment – shops – cont'd

$\alpha 1 \in \{G, J, F, O, X\}$

A→B indicates A must be completed before B start

J = Job-shop problem. It is a general shop scheduling problem with chain precedence constraints of the form $O_{1j} \rightarrow O_{2j} \rightarrow \ldots \rightarrow O_{n(j)j}$

F = Flow-shop problem. It is a special job-shop problem with $n(j) = m$ operations for $j = 1, \ldots, n$ and $\mu_{ij} = M_i$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$

O = Open shop is same as flow shop, but there are no precedence relations between the operations.

X = Mixed shop is a combination of a job shop and an open shop

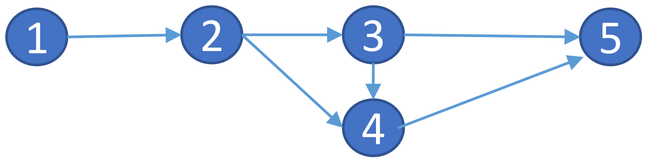
Source: Peter Brucker, Scheduling Algorithms, 2007

Job characteristics (β)

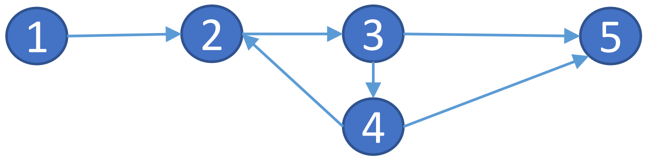
$\beta = \circ$

β_1 = pre-emption or job splitting (process interruption allowed)

β_2 = description of precedence relations (e.g. acyclic directed graph)



acyclic graph



cyclic graph

β_3 = release times (r_i) specified for each job

β_4 = specifies restrictions on the processing times or on the number of operations

β_5 = specifies if deadlines (d_i) are set for each J_i

β_6 = specifies if it is a p-batch or s-batch

β_6

- A batch is a set of jobs that can be processed jointly.
- The completion time of all the jobs in a batch is the finishing time of the last job in the batch.
- In the p-batch set of problems, the length of a batch is defined as the **maximum** processing time of any job in the batch.
- The s-batch set of problems has a different definition for the length of a batch, namely, it is partitioned into a **setup time** and the **sum of the processing times of the** jobs in the batch.

Source: J. Diaz et al. . (2006)

Optimality criteria

C_i is the finishing time of job J_i and $f_i(C_i)$ is the associated cost

Types of total cost functions

$$f_{max}(C) \coloneqq max\{f_i(C_i) | i = 1, \ldots, n\}$$ **Bottleneck** objectives

and

$$\sum f_i(C) = \sum_{i=1}^{i=n} f_i(C_i)$$ **Sum** objectives

Optimality criteria

C_i = The completion time of job J_i — $C_i \neq p_i\ \forall\ i=2:n$

F_i = The flow time of job J_i — $C_i\text{-}r_i$

L_i = Lateness of job J_i — $C_i\text{-}d_i$

T_i = Tardiness of job J_i — $max\{0, L_i\}$

E_i = Earliness of job J_i — $max\{0, -L_i\}$

$\delta_i = 1$ if job *i* is tardy ($T_i>0$)

$\delta_i = 0$ if job *i* is in time ($T_i=0$)

$$C_{max} = Max_{i=1:n}\{C_i\}$$ Makespan

$$L_{max} = Max_{i=1:n}\{L_i\}$$ Maximum lateness

$$T_{max} = Max_{i=1:n}\{T_i\}$$ Maximum tardiness

Flow and job shop

- A flow shop is an arrangement of machines, such that all the jobs visit machines in the order in which they are arranged. This is applicable to a scheduling system **where all jobs are required to visit all the machines in the same order**.

- A job shop scheduling problem is one **where each job has its own set of machines** that it visits in a given order. This is different from a flow shop where all the jobs visit all the machines in the same order.

Example – 4 jobs 3 person job shop problem

Project A					
Job number (i)	Operation (O_1)	Operation (O_2)	Operation (O_3)	Release date (r_i)	Deadline (d_i)
1	$P_{11} = 4$	$P_{12} = 3$	$P_{13} = 2$	0	16
2	$P_{22} = 1$	$P_{21} = 4$	$P_{23} = 4$	0	14
3	$P_{33} = 3$	$P_{32} = 2$	$P_{31} = 3$	0	10
4	$P_{42} = 3$	$P_{43} = 3$	$P_{41} = 1$	0	8

P_{ik} = Time to process job i by person k

Given sequence

<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th></tr>
<tr><td>M_1</td><td></td><td colspan="4">2</td><td colspan="4">1</td><td>4</td><td colspan="3">3</td><td></td><td></td><td></td></tr>
<tr><td>M_2</td><td>2</td><td colspan="3">4</td><td colspan="2">3</td><td></td><td></td><td></td><td colspan="3">1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>M_3</td><td colspan="3">3</td><td></td><td colspan="3">4</td><td colspan="4">2</td><td></td><td colspan="2">1</td><td></td><td></td></tr>
</table>

Outcome of scheduling

<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th></tr>
<tr><td>M_1</td><td></td><td colspan="4">2</td><td colspan="4">1</td><td>4</td><td colspan="3">3</td><td></td><td></td><td></td></tr>
<tr><td>M_2</td><td>2</td><td colspan="3">4</td><td colspan="2">3</td><td></td><td></td><td></td><td colspan="3">1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>M_3</td><td colspan="3">3</td><td></td><td colspan="3">4</td><td colspan="4">2</td><td></td><td colspan="2">1</td><td></td><td></td></tr>
</table>

Makespan (maximum completion time)

$$C_{max} = Max_{i=1:n}\{C_i\} = Max\{C_1, C_2, C_3, C_4\} = Max\{14,11,13,10\} = 14$$

Total flow time

$$\sum F_i = \sum C_i - r_i = 14 + 11 + 13 + 10 = 48$$

Total lateness [L_i=C_i-d_i]

$$\sum L_i = (14 - 16) + (11 - 14) + (13 - 10) + (10 - 8) = 0$$

Total tardiness $[max\{0, L_i\}]$ | L_i>0

$$\sum T_i = max\{3,2,0,0\} = 3$$

2δ = 2 tardy jobs (J_4, J_3)

Some basic sequencing rules

- **FCFS (First Come First Served)** Jobs processed in the order they come to the shop

  J_i is in order i
- **SPT (Shortest Processing Time)** Jobs with the shortest processing time are scheduled first

  $J_{[x,\, y,\, \ldots n]}$ is in order $p_x \leq p_y \leq \ldots\ p_n$
- **EDD (Earliest Due Date or Deadline)** Jobs are sequenced according to their due dates

  $J_{[x,\, y,\, \ldots n]}$ is in order $d_x \leq d_y \leq \ldots\ d_n$
- **CR (Critical Ratio)** Compute the ratio of processing time of the job and remaining time until the due date. Schedule the job with the smallest CR value next.

$$CR_i = \frac{d_i - CT_i}{p_i}$$

Current time (CT_i) is $\sum p_j$ with j representing the index of all selected p in the sequence

Source: R Lindeke, 2005

SPT proof - single machine

Shortest processing time, so we look at completion time in terms of processing time.

$$[C_1, C_2, \dots C_n] = [p_1, (p_1 + p_2), \dots, (p_1 + \cdots + p_n)]$$

$$\sum_{i=1}^{n} (n - i + 1) p_i$$

$(n - i + 1)$ ↓

p_i ↑

Aim to obtain minimum $\sum C_i$

x	y(i:n)	y(n:i)	Prod x & y(i:n)	Prod x & y(n:i)
1	1	10	1	10
2	2	9	4	18
3	3	8	9	24
4	4	7	16	28
5	5	6	25	30
6	6	5	36	30
7	7	4	49	28
8	8	3	64	24
9	9	2	81	18
10	10	1	100	10
			∑ 385	220

Single machine scheduling

i	1	2	3	4	5	6
p_i	20	13	39	41	3	1
d_i	70	81	55	21	23	30
r_i	0	0	0	0	0	0

	FCFS	SPT	EDD	CR
$F_{avg} = \frac{\sum_1^n C_i - r_i}{n}$	78.5	42	72.5	72.2
$L_{avg} = \frac{\sum_1^n C_i - di}{n}$	31.8	-4.6	25.8	25.5
# tardy jobs	4	2	6	6

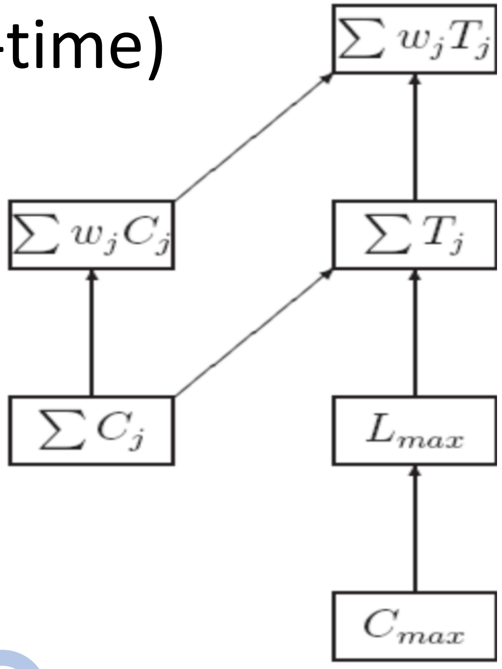
- SPT will guarantee minimum Mean Flow time
- Minimize Maximum lateness with SPT model in this case

Single machine stochastic scheduling

- Assume that p_i (processing times) are random variables and the objective is to minimize average weighted flow time, jobs are sequenced according to expected weighted SPT.
- Σ $\boldsymbol{C_j}$ (mean flow-time) becomes Σ ω_j $\boldsymbol{C_j}$ (weighted flow-time)
- Σ ω_j $\boldsymbol{T_j}$ (weighted sum of tardiness)

where the tardiness of job j is given by

$$T_j = max\ \{\ 0,\ C_j - d_j\ \}.$$



MPiE

Questions?

jeroen.bergmann@eng.ox.ac.uk