Operations Scheduling

MG2029 Production planning and control

Seyoum Eshetu Birkie
Associate Professor
Operations & production management
seyoume@kth.se

Operations Scheduling

MG2029 Production planning and control

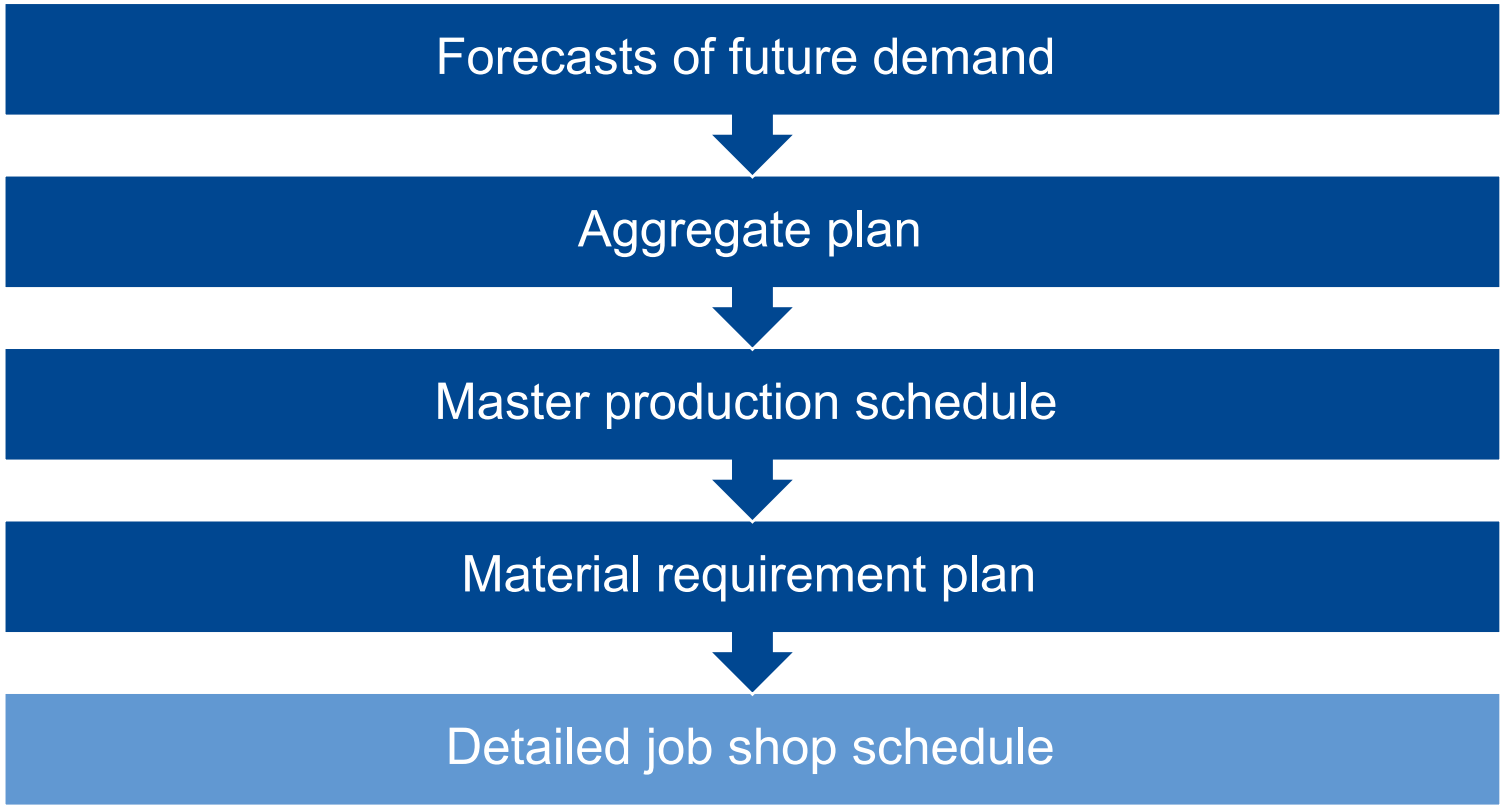
This lecture material is based on:

- ”Production and operation analysis” by Nahmias S. (7th ed) Chapter 9: sections 9.2 – 9.4, 9.6-9.7, 9.10
- Hakan Akillioglu, researcher KTH

Scheduling : Definition

- **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 (Morton and Pentico, 1993).
- **Scheduling specifies** the time each job starts and completes on each machine, as well as any additional resources needed.
- **A sequence is** a simple ordering of the jobs.

Planning hierarchy

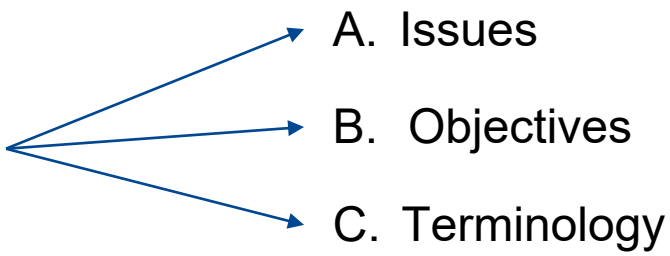


Operations scheduling is complex

Determining a **best** sequence

- 32 jobs on a single machine
- 32! possible sequences approx. $2{,}6 \times 10^{35}$
  - suppose a computer could examine one billion sequences per second
  - It would take 8.4×10^{15} centuries to evaluate all sequencing options
- Real life problems are much more complicated!
- Scheduling theory helps to
  - classify the problems
  - identify appropriate measures
  - develop solution procedures

Characteristics of operations scheduling



A. Issues

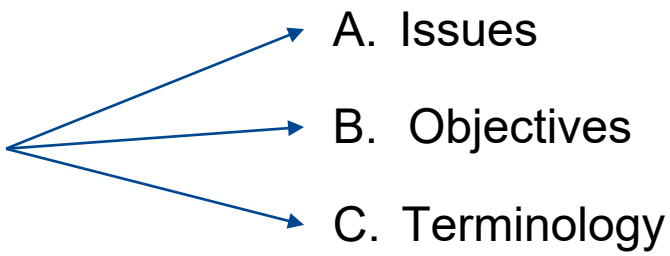
- ❑ Job arrival (loading) pattern
  - ❑ Scheduling is not static, it is dynamic in real.
- ❑ Number and types of machines
- ❑ Number of workers in the shop
- ❑ Material flow and sequencing
- ❑ Evaluation of alternatives

B. Objectives

- Meet due dates
- Minimize work-in-process (WIP) inventory
- Minimize average flow time
- Maximize machine/worker utilization
- Reduce set-up times for changeovers
- Minimize direct production and labor costs

(note: that these objectives can be conflicting)

Characteristics of operations scheduling

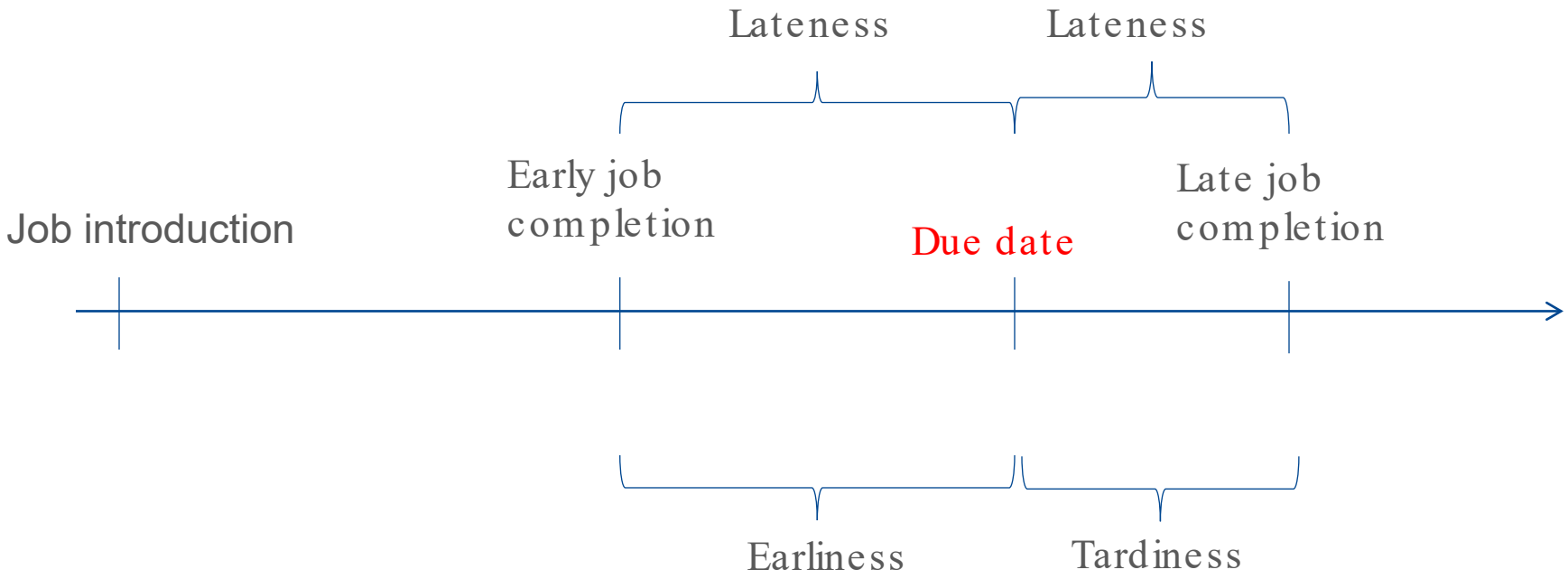


C. Terminology

- Flow shop: n jobs processed through m machines in the same sequence
- Job shop: the sequencing of jobs through machines may be different, and there may be multiple operations on some machines.
- Parallel processing vs. sequential processing: parallel processing means that the machines are identical.
- Flow time of job i: Time elapsed from initiation of first job until completion of job i.
- Makespan: Flow time of the job completed last. (i.e. total completion time of the whole batch.)
- Tardiness: The positive difference between the completion time and the due date.
- Lateness: Difference between completion time and due date (can be negative).
- Slack time: time from now till the due date minus the total processing time of the job.

Characteristics of operations scheduling

A. Issues
B. Objectives
C. Terminology



Basic sequencing rules

- **FCFS** First Come First Served
  - Jobs are processed in the order they come to the shop
- **SPT** Shortest Processing Time
  - Jobs with the shortest processing time are scheduled first
- **EDD** Earliest Due Date
  - Jobs are sequenced according to their due dates
- **CR** Critical Ratio
  - Compute the ratio of remaining time until the due date and processing time of the job
  - Schedule the job with the smallest CR value next
- **S/OP** Minimum slack time per operation
  - Choose the next job on the basis of which job has the least slack time
  - To standardize or normalize, the slack time is divided by the number of operations

Basic sequencing rules: an example

- Suppose that 5 jobs will be processed on a single machine. The jobs are ready for processing at time t=0.

Job	Processing time	Due date	Order of arrival
A	5	10	1
B	10	15	2
C	2	5	3
D	8	12	4
E	6	8	5

Basic sequencing rules - FCFS

Job	Processing time	Due date	Completion time	Tardiness
A	5	10	5	0
B	10	15	15	0
C	2	5	17	12
D	8	12	25	13
E	6	8	31	23
Totals			**93**	**48**

Mean flow time:	93/5	=18,6
Average tardiness :	48/5	=9.6
Number of tardy jobs:		=3
Max. Tardiness:		=23

Basic sequencing rules - SPT

Job	Processing time	Due date	Completion time	Tardiness
C	2	5	2	0
A	5	10	7	0
E	6	8	13	5
D	8	12	21	9
B	10	15	31	16
Totals			**74**	**30**

Mean flow time:	74/5	=14,8
Average tardiness :	30/5	=6
Number of tardy jobs:		=3
Max. Tardiness:		=16

Basic sequencing rules - EDD

Job	Processing time	Due date	Completion time	Tardiness
C	2	5	2	0
E	6	8	8	0
A	5	10	13	3
D	8	12	21	9
B	10	15	31	16
Totals			**75**	**28**

Mean flow time:	75/5	=15
Average tardiness :	28/5	=5,6
Number of tardy jobs:		=3
Max. Tardiness:		=16

Basic sequencing rules - CR

- **Critical Ratio**, CR = time remaining / work remaining

$$= \frac{Due\ date\ - Now}{Remaining\ processing\ time}$$

- Each time a job is scheduled, CR is recalculated for every unscheduled job.
- CR selection criteria
  - Jobs with the smallest CR are run first.
  - Jobs with negative CR are scheduled first.
  - If there is more than one job with negative CR, then those jobs are sequenced in SPT order.

Basic sequencing rules - CR

Step 1
t=0

Job	Processing time	Due date	Critical ratio
A	5	10	10/5 = 2
B	10	15	15/10 = 1,5
C	2	5	5/2 = 2,5
D	8	12	12/8 = 1,5
E	6	8	8/6 = 1,33

Step 2
t=6

Job	Processing time	Due date	Critical ratio
A	5	4	4/5 = 0,8
B	10	9	9/10 = 0,9
C	2	-1	-1/2 = -0,5
D	8	6	6/8 = 0,66

Step 3
t=8

Job	Processing time	Due date	Critical ratio
A	5	2	2/5 = 0,4
B	10	7	7/10 = 0,7
D	8	4	4/8 = 0,5

Step 4
t=13

Job	Processing time	Due date	Critical ratio
B	10	2	2/10 = 0,2
D	8	-1	-1/8 = 0,12

Sequence: E – C – A – D – B

Basic sequencing rules - CR

Job	Processing time	Due date	Completion time	Tardiness
E	6	8	6	0
C	2	5	8	3
A	5	10	13	3
D	8	12	21	9
B	10	15	31	16
Totals			**79**	**31**

Mean flow time:	79/5	=15,8
Average tardiness :	31/5	=6,2
Number of tardy jobs:		=4
Max. Tardiness:		=16

Basic sequencing rules – S/RO

Sequence with minimum slack time per operation

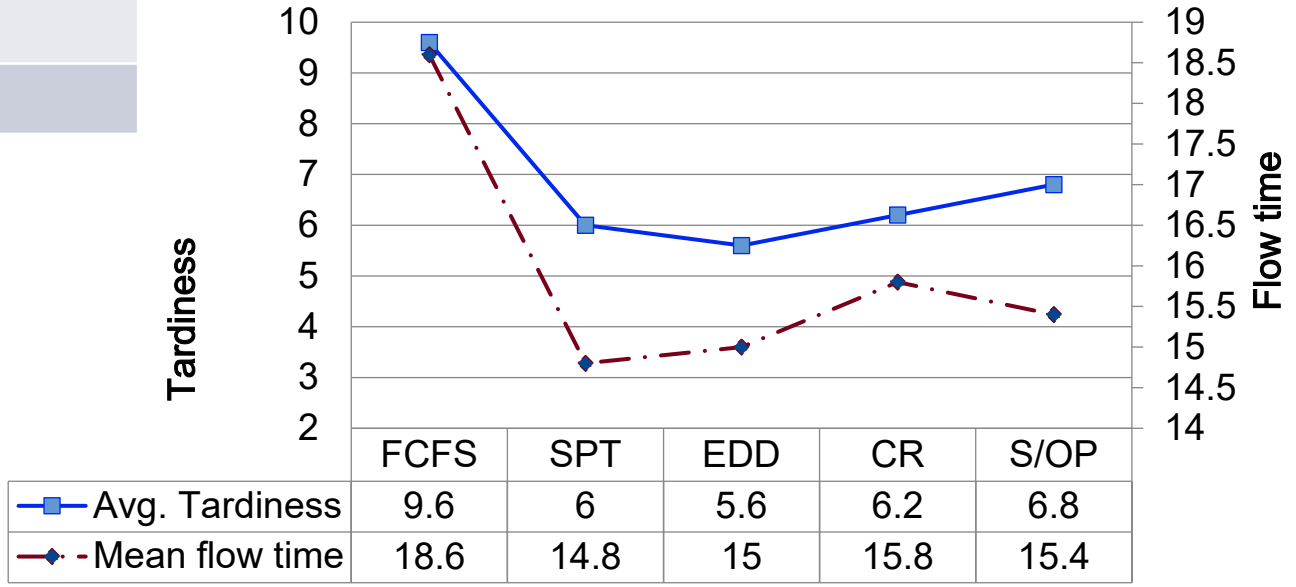
Job	Processing time	Due date	Slack	Completion time	Tardiness
E	6	8	2	6	0
C	2	5	3	8	3
D	8	12	4	16	4
A	5	10	5	21	11
B	10	15	5	31	16
Totals				**82**	**34**

Mean flow time:	82/5	=15,4
Average tardiness :	34/5	=6,8
Number of tardy jobs:		=4
Max. Tardiness:		=16

* Number of operation is assumed to be one. There is one single machine

Basic sequencing rules – a comparison

	Mean flow time	Avg. Tardiness	# of tardy jobs	Max. tardiness
FCFS	18,6	9,6	3	23
SPT	14,8	6,0	3	16
EDD	15,0	5,6	3	16
CR	15,8	6,2	4	16
S/OP	15,4	6,8	4	16



Basic sequencing rules: Remarks

- Observe that the makespan is the same for every schedule. This is expected for a single machine problem if ready times are all zero (static scheduling). For a multi-machine problem, makespan may be different from one schedule to another.
- Total completion time and mean completion time are equivalent objectives. Since mean completion time is obtained from the total completion time by dividing the total completion time by the number of jobs
  - If a schedule minimizes total completion time, it also minimizes mean completion time.

Basic sequencing rules: Remarks

- The Earliest Due Date (EDD) rule minimizes maximum lateness and maximum tardiness (for the single machine static scheduling problem).

- Shortest processing time (SPT) rule minimizes total completion time, mean flow time and total lateness, mean lateness (for the single machine static scheduling problem).

Moore's algorithm

Algorithm for minimizing number of tardy jobs

Step 1. Sequence jobs according to EDD for initial solution.

Step 2. Find the first tardy job in the current sequence, say job [i]. If none exists, go to step 4.

Step 3. Consider jobs [1],[2],...,i. Reject the job with the largest processing time. Return to step 2.

Step 4. Form an optimal sequence by taking the current sequence and appending to it the rejected jobs. The jobs appended to the current sequence may be scheduled in any order because they constitute the tardy jobs.

Moore's algorithm

Algorithm for minimizing the number of tardy jobs

Say there 5 jobs, as follows.

Job	Processing time	Due date
A	7	9
B	8	17
C	4	18
D	6	19
E	6	21

Arrange the jobs in the EDD order and find if any is tardy

Job	Processing time	Due date	Completion time
A	7	9	7
B	8	17	15
C	4	18	19
D	6	19	25
E	6	21	31

First tardy job

Moore's algorithm

Eliminate the job with the longest processing time among the jobs before the first tardy job.

Eliminate the job with longest processing time

Job	Processing time	Due date	Completion time
A	7	9	7
B	8	17	15
C	4	18	19

Now, jobs to consider

Job	Processing time	Due date	Completion time
A	7	9	7
C	4	18	11
D	6	19	17
E	6	21	23

First tardy job

Moore's algorithm

- Eliminate A, since it has the longest processing time.

Job	Processing time	Due date	Completion time
A	7	9	7
C	4	18	11
D	6	19	17
E	6	21	23

Now, jobs to consider

Job	Processing time	Due date	Completion time
C	4	18	4
D	6	19	10
E	6	21	16

No tardy jobs. We are done !

- Optimal sequence with minimum number of tardy jobs is
- C-D-E-A-B or C-D-E-B-A
- The sequence of eliminated jobs does not change the result, since they are the tardy ones.

Lawler's algorithm

Sequencing with precedence constraints

- The algorithm is applicable for single machine problems with the objective of minimizing
  - Makespan
  - Maximum lateness
  - Maximum tardiness

- Consider a single machine problem with precedence constraints and minimizing maximum lateness objective (other objectives previously stated may be minimized similarly)
- General scheme: The algorithm first assigns a job to the last position, then a job to the position next to last, and so on
- Candidate job for a position: Due to precedence constraints, not all the jobs are candidates for a position
- For example, if a job has a successor, the job cannot be assigned to the last position. Hence, candidates for the last position are the ones without any successor

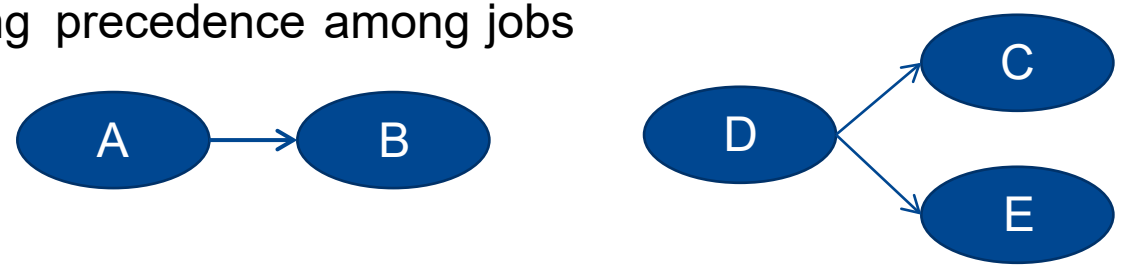
Lawler's algorithm

Which job to assign?

1. Eliminate all the jobs which are previously assigned (to later positions)
2. Identify the candidates - jobs that have no successor or have successors all previously assigned (to later positions)
3. Among all the candidates, schedule the one with the minimum lateness. (if you are targeting to minimize maximum tardiness, choose the minimum tardiness)

Lawler's algorithm - example

Consider following precedence among jobs



Job	Processing time	Due date
A	7	9
B	8	17
C	4	18
D	6	19
E	6	21

Lawler's algorithm

- Candidate jobs = {B, C, E}
- Completion time of unasssigned jobs, $\tau = 7+8+4+6+6=31$

Job	Processing time	Due date	Lateness
B	8	17	$31-17=14$
C	4	18	$31-18=13$
E	6	21	$31-21=10$

Min lateness

				E

Lawler's algorithm

- Candidate jobs = {B, C}
- $\tau = 31 - 6 = 25$

Job	Processing time	Due date	Lateness
B	8	17	25 - 17 = 8
C	4	18	25 - 18 = 7

Min lateness

			C	E

Lawler's algorithm

- Candidate jobs = {B, D}
- $\tau = 25-4=21$

Job	Processing time	Due date	Lateness
B	8	17	21-17=4
D	6	19	21-19=2

Min lateness

Lawler's algorithm

Solution turns out to be ;

A	B	D	C	E

Remember; the algorithm is described in the context of minimizing maximum lateness. To get the algorithm for minimizing maximum tardiness, replace "lateness" by "tardiness" in calculations.

Sequencing for multiple machines

Two Machine Flow Shop

- Until now, there were only one machine.
- The main characteristics of a two-machine flow shop system
  - Every job first visits Machine 1 and then Machine 2
  - Every machine can process one job at a time
  - Every job can be processed by one machine at a time
- Objective is to minimize the makespan or max flow time

Two-Machine Flow Shop

Johnson's Rule

1. List time required to process each job at each machine. Set up a one-dimensional matrix to represent desired sequence with # of slots equal to # of jobs.
2. Select smallest processing time at either machine.

If that time is on machine 1, put the job as near to beginning of sequence as possible.

If smallest time occurs on machine 2, put the job as near to the end of the sequence as possible.

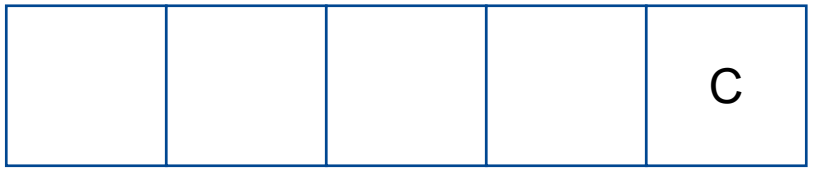
3. Remove the job from the list.
4. Repeat steps 2-3 until all slots in matrix are filled & all jobs are sequenced.

Two-Machine Flow Shop

Johnson's Rule - Example

Job	Machine center 1	Machine center 2
A	5	6
B	16	5
~~C~~	~~8~~	~~2~~
D	9	17
E	4	6

The minimum processing time, 2, is given by Job C on Machine 2. So, Schedule Job C in the end.



Two-Machine Flow Shop

Johnson's Rule - Example

Job	Machine center 1	Machine center 2
A	5	6
B	16	5
D	9	17
~~E~~	~~4~~	~~6~~

Job	Machine center 1	Machine center 2
~~A~~	~~5~~	~~6~~
B	16	5
D	9	17

Then Job B is placed to the end.

The minimum processing time, 4, is given by Job E on Machine 1. So, Schedule Job E in the beginning.

Job E is removed. Now, there is a tie. Minimum processing time is given by Jobs A and B. Break ties arbitrarily. Schedule one of Job A or Job B.

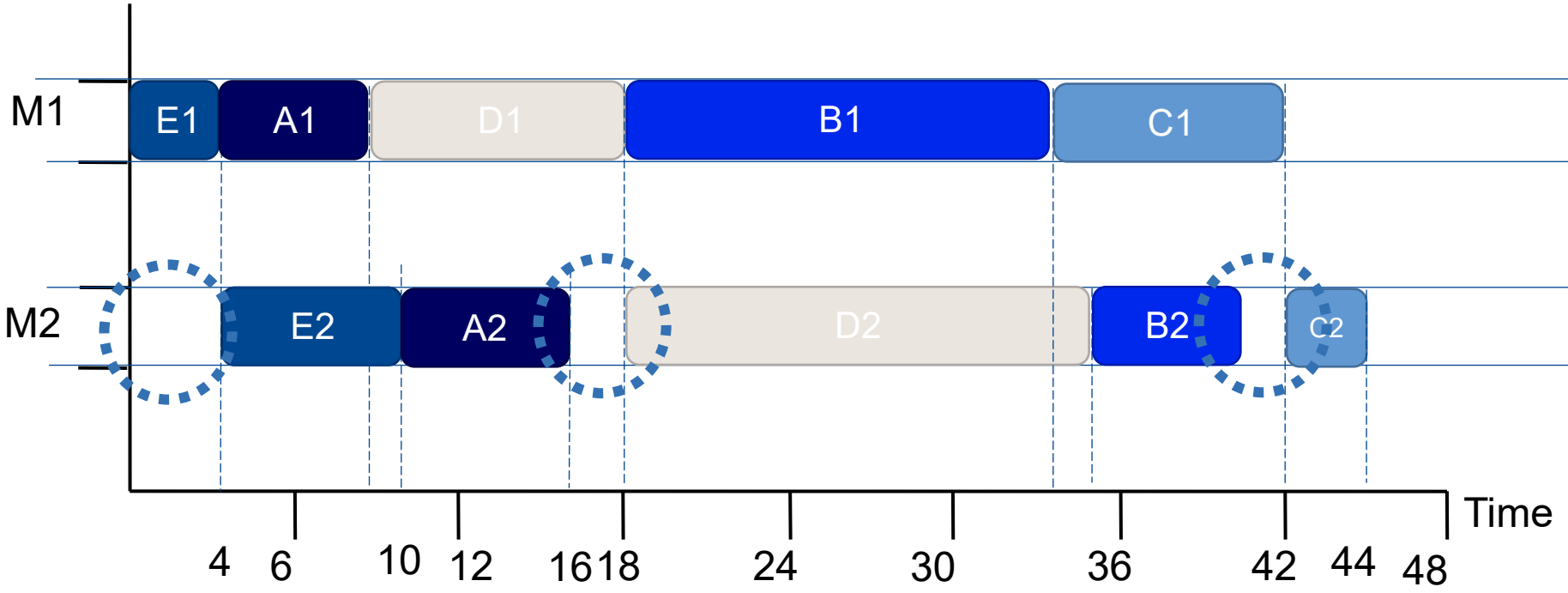
Two-Machine Flow Shop

Johnson's Rule - Example

The sequencing is complete after assigning the remaining Job D to the remaining position .

E	A	D	B	C

Johnson's rule guarantees that the above schedule gives the **best value (44) of makespan.**



Extension of Johnson's rule to 3 machines

The extension of Johnson's rule does not guarantee an optimal makespan for all three-machine flow shop cases. However, the extension guarantees an optimal makespan if the following condition is satisfied;

$$\min A_i \geq \max B_i \text{ or } \min C_i \geq \max B_i$$

Where A, B and C represent machines.

If either of those conditions is satistified, problem can be reduced to two machine case as such;

Say, $A'_i = A_i + B_i$ and $B'_i = B_i + C_i$. Now same procedure is applied for A'_i and B'_i.

Extension of Johnson's rule to 3 machines

Job	Machine A	Machine B	Machine C
1	17	12	15
2	11	5	27
3	25	9	14
4	15	7	15

$\min A_i = 11 \leq 12 = \max B_i$

$\min C_i = 14 \geqslant 12 = \max B_i$

Condition is satisfied due to the second inequality!

$A'_i = A_i + B_i$ and $B'_i = B_i + C_i$

Job	Machine A'	Machine B'
1	29	27
2	16	32
3	34	23
4	22	22

Now follow regular Johnson's algorithm as in the previous example with machine A' and B'.

Assembly Line Balancing

Characteristics

- If the demand of a standard product is sufficiently high and stable over a long period of time (as it is in a make-to-stock or assemble-to-stock production system), it's usually cost effective to rearrange resources in the order in which the resources are used.
- The job is split into many work elements. The work elements are so small that each element or a group of elements can be performed by one workstation
- The line balancing problem is to equalize work at each workstation. So, the solution procedure attempts to evenly distribute the work elements to the workstations.

Assembly Line Balancing

Characteristics

- n jobs to be processed in an assembly line
- t_i is the time for each job,
- Tasks are assigned to stations. Tasks must be sequenced properly, and certain tasks may not be done at the same station.
- The objective is to assign tasks to stations for
  - minimizing the cycle time, C, for a given number of stations,
  - minimizing the number of stations for a given cycle time.
- The general problem is difficult to solve optimally, but effective heuristics are available. (ranked positional weight technique.)

Assembly Line Balancing

Cycle time

- Cycle time: Time interval between parts coming off the line
- Required cycle time is defined by demand rate for product

Example: A manufacturer of communications equipment is constructing a line to assemble several similar models of speaker phones. An industrial engineer had divided assembly of each model into elemental tasks. Phones require about 30 operations. Task times vary from 6 to 36 seconds. What is the appropriate cycle time if demand requires producing 720 phones per shift. Each shift has 8 productive hours.

Assembly Line Balancing

Total work content:

$$T = \sum_{i=1}^{n} t_i$$

Theoritically, minimum number of workstations required for a cycle time, C is that;

$$\left[\frac{T}{C}\right]$$

(rounded up to the next integer).

Assembly Line Balancing

Ranked Positional Weight Heuristic

- Calculate the positional weight for each task in the precedence graph
- Positional weight, **PW(i): sum of processing times of the task and all other tasks following it in the graph**
- Rank the tasks in decreasing order of positional weights
- Assign tasks to stations in the order of this ranked positional weight (RPW) list, avoiding precedence constraint and cycle time violations.

Assembly Line Balancing

Ranked Positional Weight Heuristic -Example

Produce 450 units in each 7,5 hour shift

Task	Description	Task time (seconds)	Immediate predecessors
a	Mount bracket to baseboard	10	-
b	Mount baseboard to frame	45	a
c	Insert two -way speaker	30	a
d	Insert and test circuit card	25	b
e	Connect power cables	25	c,d
f	Test assembly	20	e
g	Enclose and seal	20	d,f
h	Pack	10	g

Assembly Line Balancing

Ranked Positional Weight Heuristic -Example

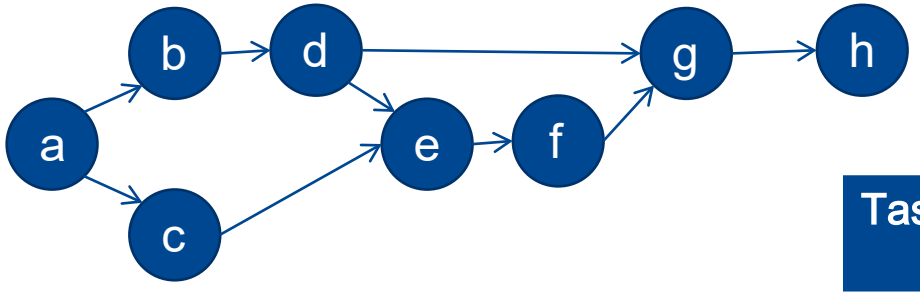
Cycle time; $\frac{7{,}5*60*60}{450} = 60\ secs$

Minimum number of stations required=

$$\left[\frac{T}{C}\right] = \left[\frac{\sum_{i=1}^{n} t_i}{C}\right] = [\frac{185}{60}] = 4$$

Assembly Line Balancing

Precedence diagram



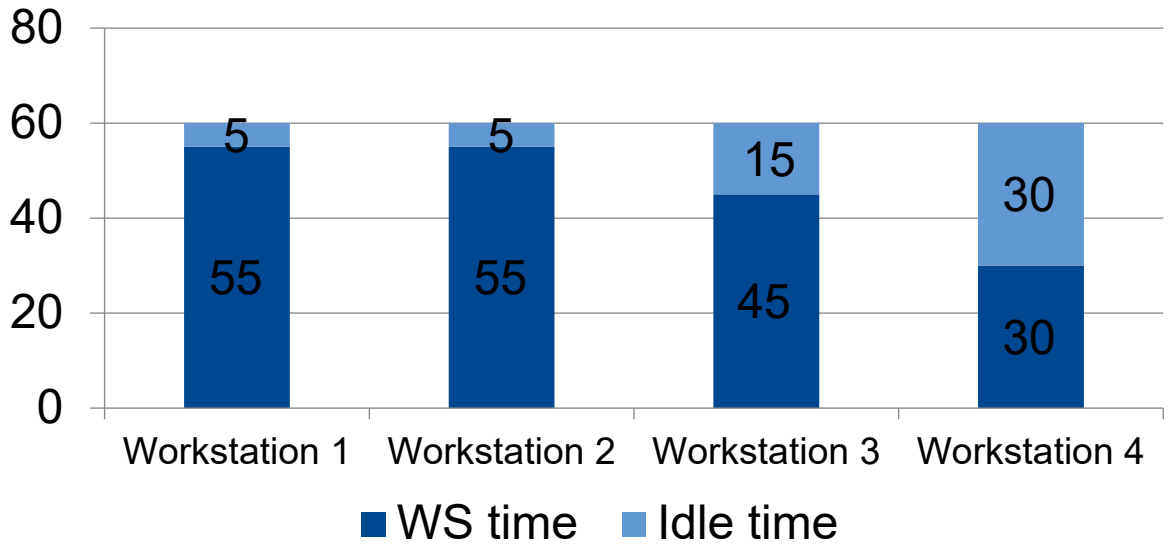
Task	Task time (seconds)	Positional weights
a	10	185
b	45	145
c	30	105
d	25	100
e	25	75
f	20	50
g	20	30
h	10	10

Already ranked!

Assembly Line Balancing

Ranked Positional Weight Heuristic -Example

Station	1	2	3	4
Tasks	a,b	c,d	e,f	g,h
Idle time	5	5	15	30




KTH
VETENSKAP
OCH KONST