Scheduling

Notes -4

Example
Compare SPT, EDD and Moore’s Algorithm

Job	Processing Time	Due date
1	6	18
2	2	6
3	3	9
4	4	11
5	5	8

Solution

Method	Mean Flow Time	Average Tardiness	Number of Tardy Jobs	T_{max}
SPT	10	1.6	2	6
EDD	10.6	1.2	3	3
Moore	10.2	2.4	1	12

Smith’s Algorithm

- If some condition is sufficient but not necessary for optimizing a given measure of performance, then there may be many different optimal schedules. We may then try to optimize some ***secondary measure of performance*** over this set of solutions, although the solution may not be optimal for this secondary measure over all possible schedules.
- EDD was sufficient, to minimize maximum tardiness.
- SPT was both necessary and sufficient to minimize mean completion time.

Smith's Algorithm

- Smith's rule (1956) may be used to minimize mean completion time over all schedules that have minimal maximum tardiness T_{max}
- Step 1: K=n , t =$\sum_{i=1}^{n} P_i$ and U ={J1, J2,…..,Jn}
- Step 2: Find the jobs
  - di ≥t and
  - that has maximum processing time over all jobs
- Step 3: K= n-1 and the processing time of the scheduled job is subtracted from total completion time.
- Step 4: When all the jobs are scheduled, the algorithm finishes.

Example
According to Tmax =0 , solve the $4/1//\bar{F}$

Job	J1	J2	J3	J4
Pi	2	3	1	2
di	5	6	7	8

- K=4 and t=8   U={J1, J2, J3,J4}
- J4 meets the condition and J4 is selected and sequenced at the last position.
- _ - _ - _- 4

- K=3   t=6   and U={J1,J2,J3}
- J2 and J3 meet the conditions. Processing time of J2 is greater, so J2 is selected.
- _- _- 2- 4

- K=2   t=3   U={J1, J3}

J1 and J3 meet conditions. J1 is selected.

_ -1- 2 -4

K=1   t = 1 U ={J3}

Optimal sequence : 3- 1 - 2- 4

$\overline{F}$ **= 18/4**

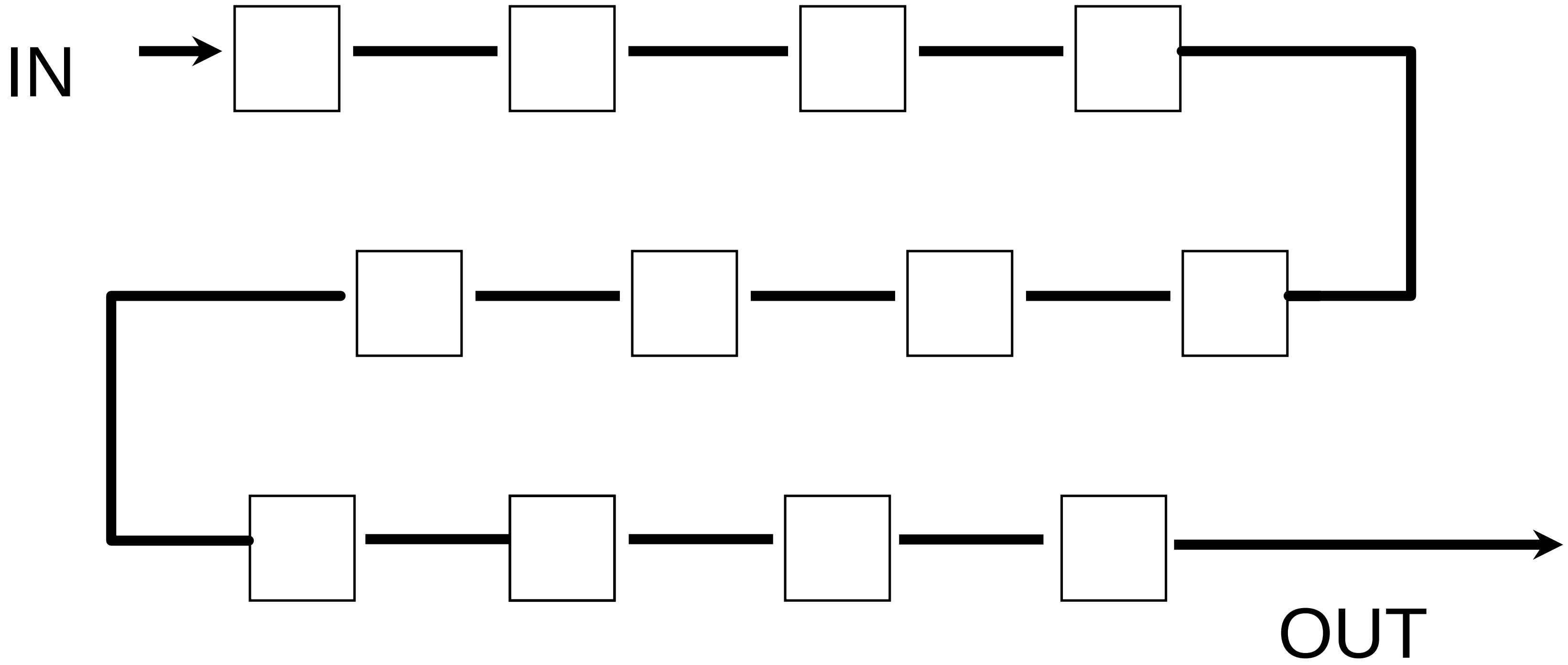
Flow Shops

- In many manufacturing and assembly facilities each job has to undergo a series of operations. Often, these operations have to be done on all jobs in the same order implying that the jobs have to follow the same route. The machines are then assumed to be set up in series and the environment is referred to as a flow shop.
- Jobs processed sequentially on multiple machines.
- All jobs are processed in the same order.

Flow Shops

  - High volume of production: so, it’s feasible to make a high capital investment
  - Less variation: so, there is less uncertainty
  - Standard products: so, there is a predictable pattern of flow of jobs through the machines.
  - The production system is not flexible; it cannot produce items if design changes significantly.
- A flow shop is suitable for a make-to-stock or assemble-to-stock production system where standard products are produced in high volume.

Flow Shops



- Customizing and painting
- Machining and polishing
- Repair and testing

Multiple Machine Flow Shop

- N jobs on M machines
- $(N!)^M$ possible sequences
- We will analyze two and three machine systems.

We assume that

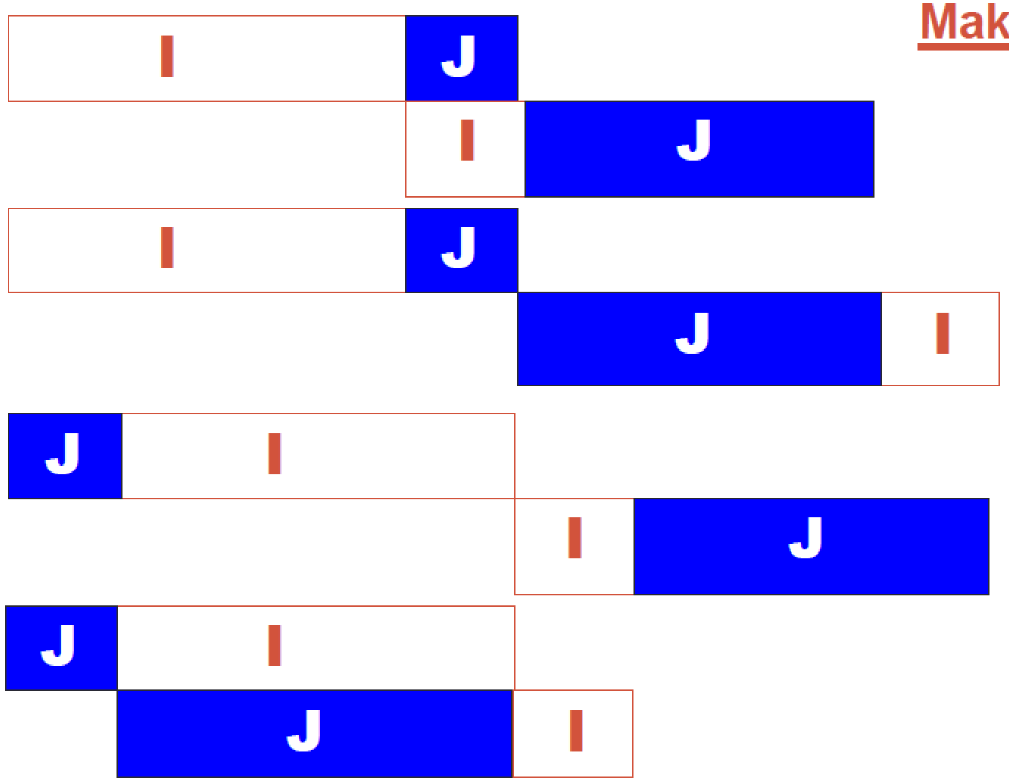
- every machine can process one job at a time.
- every job can be processed by one machine at a time

- 2 jobs, 2 machines.
- Job M1 M2
- I 4 1
- J 1 4
- There will be four possible sequences:

	M1	M2
1	I J	I J
2	I J	J I
3	J I	J I
4	J I	I J

Two Machines

Sequence	Makespan
IJ IJ	9
IJ JI	10
JI IJ	10
JI JI	6



The same order on both machines is optimal. So, if Machine 1 processes Job 1 before Job 2, then Machine 2 will also process Job 1 before Job 2.

When the job-order is the same on all machines, the schedule is called a permutation schedule.

Johnson's Algorithm

- n/2/F/Cmax or n/2/F/Fmax
- Johnson's rule is a procedure that minimizes makespan when scheduling a group of jobs on two machines.
- In the scheduling of two or more machines in a flow shop, the makespan varies according to the sequence chosen. Determining a production sequence for a group of jobs to minimize the makespan has two advantages:
- 1. The group of jobs is completed in minimum time.
- 2. The utilization of the two-machines flow shop is maximized. Utilizing the first machine continuously until it processes the last job minimizes the idle time on the *second* machine.

Johnson’s Algorithm

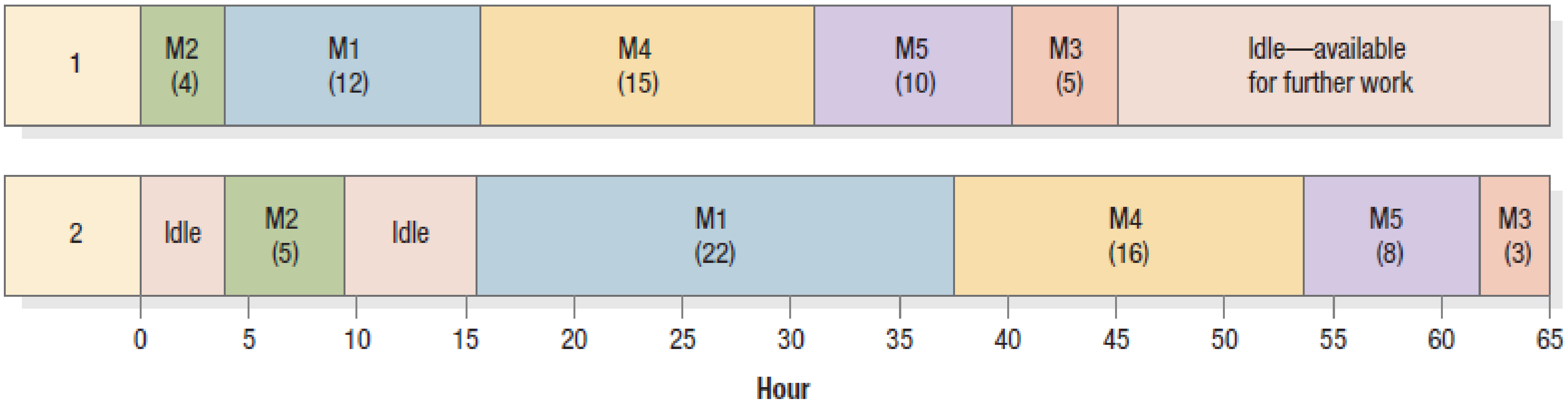
- Step 1
  - » Select the job with the lowest processing time from the schedulable job list
  - » If the list is empty, the procedure is finished
  - » If the processing time of the job selected is from machine 1, go to step 2, otherwise, go to step 3
- Step 2
  - » Schedule the job in the earliest position of the sequence and remove it from the schedulable job list
  - » Return to step 1
- Step 3
  - » Schedule the job in the latest position of the sequence and remove it from the schedulable job list
  - » Return to step 1

Example: Find the optimal order and Cmax

Jobs	Machine 1	Machine 2
M1	12	22
M2	4	5
M3	5	3
M4	15	16
M5	10	8

Optimal Order : 2 - 1- 4- 5- 3

Example



1
M2
(4)
M1
(12)
M4
(15)
M5
(10)
M3
(5)
Idle—available
for further work
2
Idle
M2
(5)
Idle
M1
(22)
M4
(16)
M5
(8)
M3
(3)
0
5
10
15
20
25
30
35
40
45
50
55
60
65
Hour

Resources

- Sıralama ve Programlama, Hüseyin Başlıgil
- Çizelgeleme Ders Notları, Prof. Dr. Hüseyin Başlıgil
- Scheduling, Theory, Algorithms and Systems, Michael L. Pinedo, Third Edition, Springer, 2008.
- https://fenix.tecnico.ulisboa.pt/downloadFile/282093452004307/5.1%20-%20Scheduling.pdf
- web4.uwindsor.ca/users/b/baki%20fazle/Chapter_08_Lecture_12_to_19_w08_431_scheduling.ppt – Windsor University Operations Scheduling Lecture Notes
- http://wps.prenhall.com/wps/media/objects/7117/7288732/krm9e_SuppJ.pdf
- http://business.unr.edu/faculty/ronlembke/352/ppt/16-scheduling.pdf
- http://rbutterworth.nfshost.com/Scheduling/1.3-due
- http://www.egyankosh.ac.in/bitstream/123456789/20794/1/Unit-7.pdf