SMART
INDUSTRY
LABORATORY

Scheduling Algorithms (7)

- Production Scheduling Algorithm (1) -

Graduate School of Information,
Production and Systems

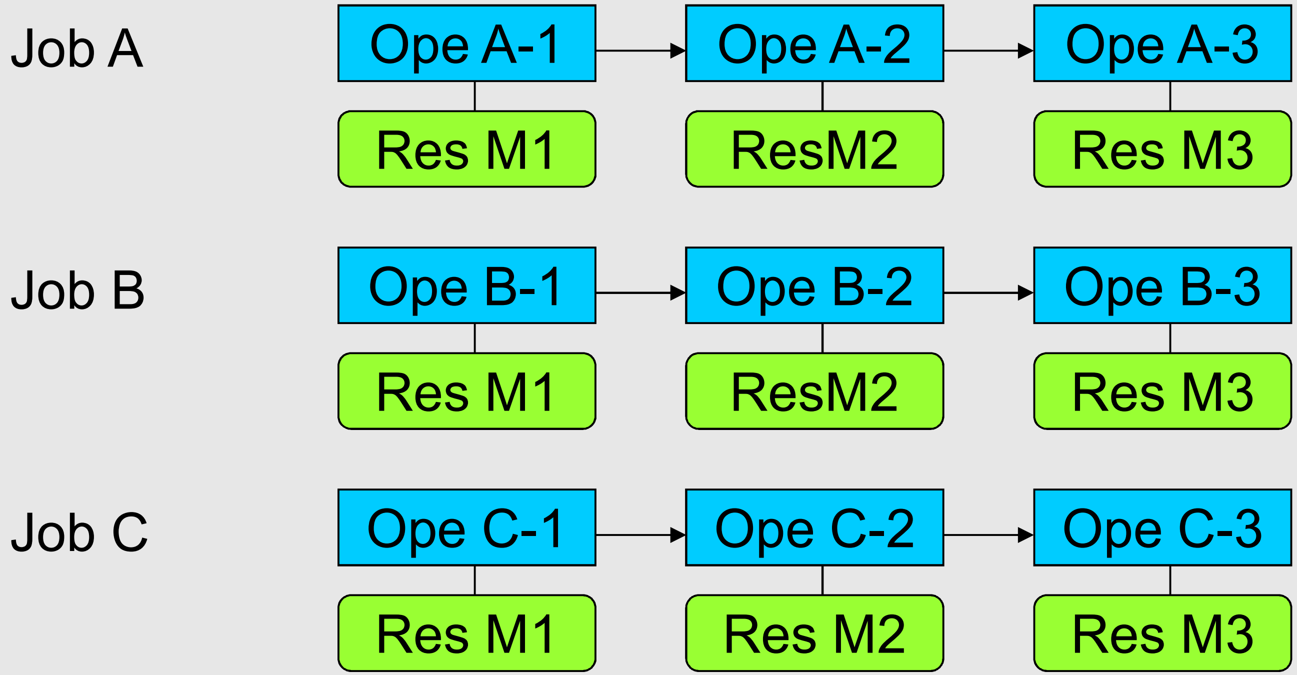
Shigeru FUJIMURA

Production Scheduling Problems

- There are several ***jobs***.
- A ***job*** consists of ***operations***.
- One ***job*** executes ***operations*** in series.
- One ***operation*** uses a ***resource*** in it's execution.

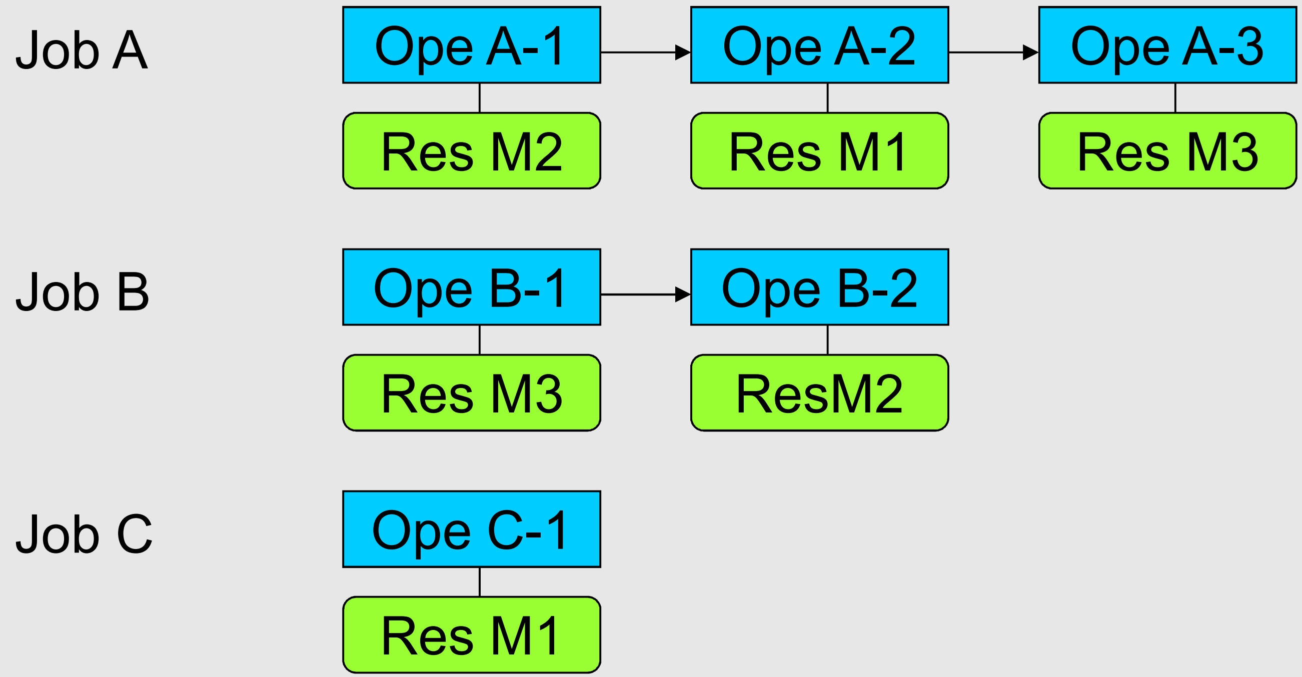
Flow Shop Scheduling Problems: FSSP

- ***Jobs*** have the same number of ***operations***.
- ***Operation's resources*** are used in the same sequence for all ***jobs***.



Job Shop Scheduling Problems: JSSP

- Numbers of ***Operations*** are not same for all ***jobs***.
- Sequence of ***operation's resources*** is not same for all ***jobs***.



Problem Definition of JSSP

Minimize C_{max}

Subject to

$C_{max} >= s_{j,n_j} + \mathbf{p}_{j,n_j}$ $\quad j=1, 2, \ldots, \mathbf{J}$

$s_{j,i} >= 0$ $\quad j=1, 2, \ldots, \mathbf{J},\ i=1, 2, \ldots, \mathbf{n}_j$

$s_{j,i+1} >= s_{j,i} + \mathbf{p}_{j,i}$ $\quad j=1, 2, \ldots, \mathbf{J},\ i=1, 2, \ldots, \mathbf{n}_j - 1$

$s_{j,i} >= s_{l,k} + \mathbf{p}_{l,k} - \mathbf{M}*(1-x_{j,i,l,k})$

$s_{l,k} >= s_{j,i} + \mathbf{p}_{j,i} - \mathbf{M}*x_{j,i,l,k}$

$j \neq l,\ j,l = 1, 2, \ldots, \mathbf{J},\ i = 1, 2, \ldots, \mathbf{n}_j,\ k = 1, 2, \ldots, \mathbf{n}_l$

$\mathbf{m}_{j,i} = \mathbf{m}_{l,k}$

Fixed values

- **J**: number of Jobs
- $\mathbf{n}_j$: number of operations of job j
- $\mathbf{p}_{j,i}$: processing time of operation i of job j
- $\mathbf{m}_{j,i}$: machine used by operation i of job j

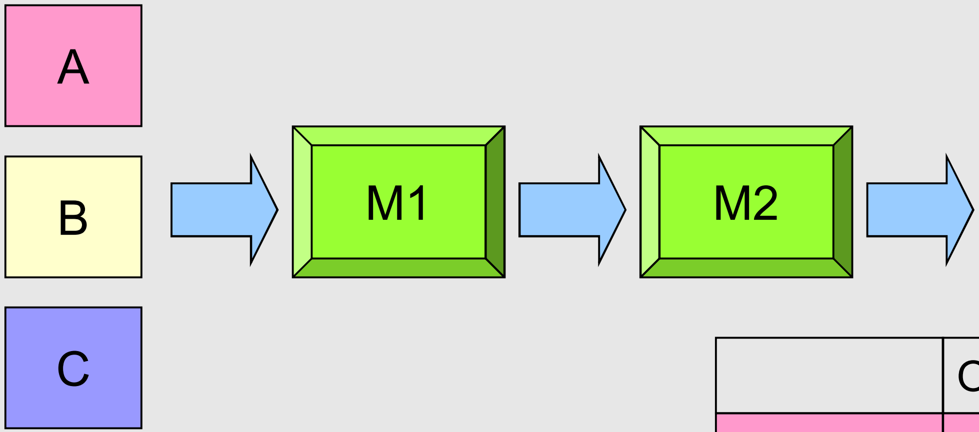
Decision Variables

- C_{max} : makespan
- $s_{j,i}$: starting time of operation i of job j
- $x_{j,i,l,k}$: 1 or 0

Scheduling Algorithms

There are 3 jobs using M1 and M2.

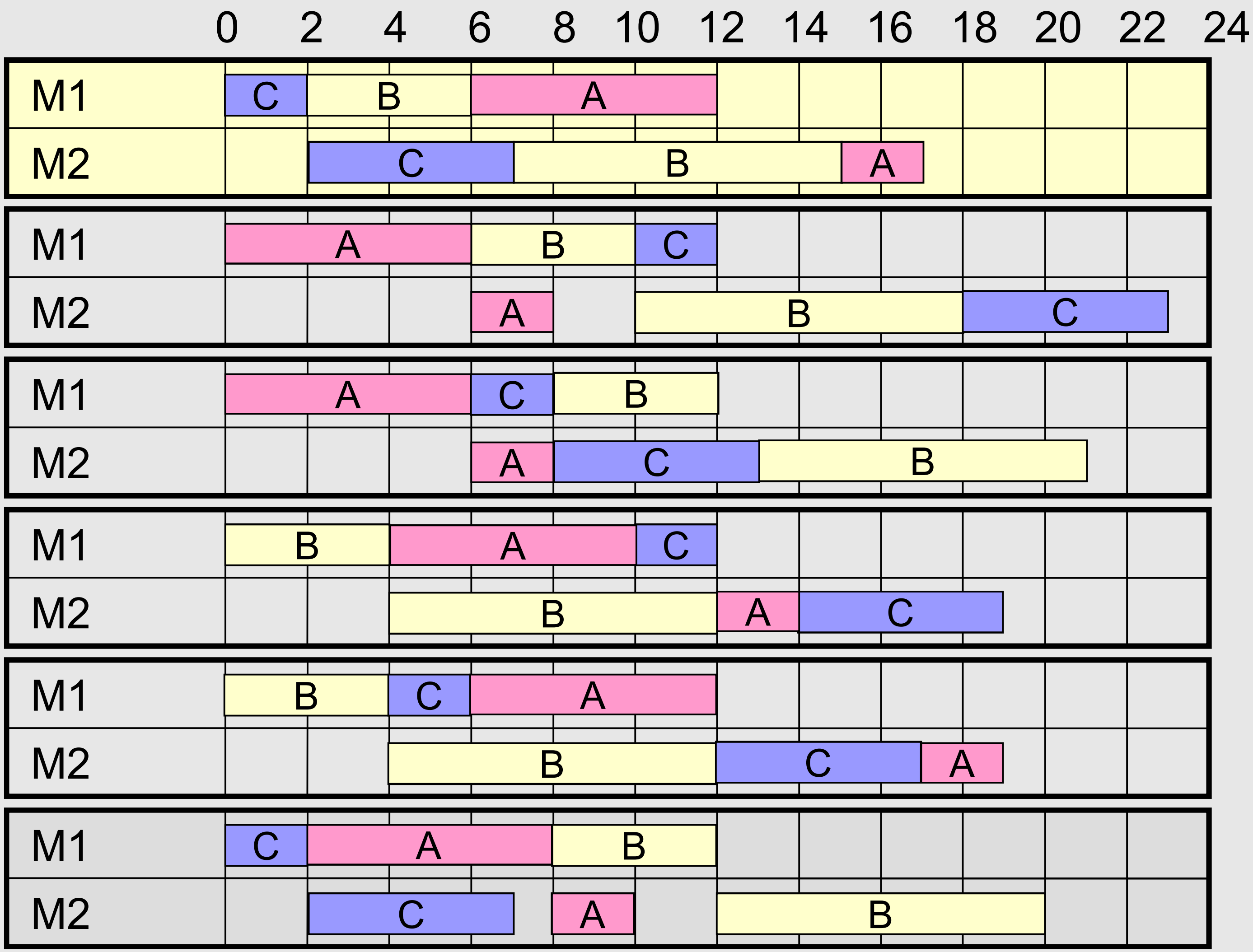
Derive production order of jobs
such that the total processing time (makespan) becomes the shortest.



	Ope1 / M1	Ope2 / M2
Job A	6h	2h
Job B	4h	8h
Job C	2h	5h

Scheduling Results

The best production order: C⇒B⇒A



Johnson Algorithm

Johnson Algorithm is an optimization method for FSSP with two operations.

Step0: All operations are set into ***Operation Set***.

Step1: Pick an operation which processing time is the shortest in ***Operation Set***. (If some operations have the same shortest time, it is not matter which operation is picked.)

Step2: In a case that the picked operation is the first (or second) operation in a job, it's job is added at the end of ***Earlier Item Sequence*** (or at the first of ***Later Item Sequence***). pick out the second (or first) operation of the job from ***Operation Set***.

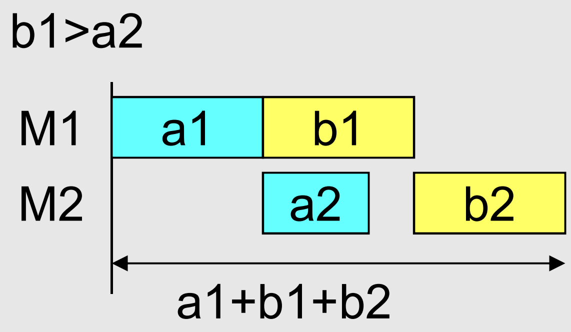
Step3: If the list of remaining operations is empty, concatenate the earlier item sequence and the later item sequence. If it is not empty, go to Step1.

Earlier Item Sequence ***Later Item Sequence***

	Ope1 / M1	Ope2 / M2
Job A	6h	2h
Job B	4h	8h
Job C	2h	5h

Proof of Johnson Algorithm

Job	M1	M2
A	a1	a2
B	b1	b2

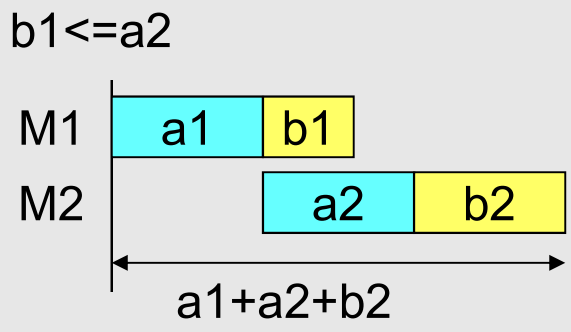


Total Lead Time for A→B

$= a1+b1+b2 \quad (b1>a2)$

$\quad a1+a2+b2 \quad (b1<=a2)$

$= a1+b2+\max\{b1,a2\}$



Proof of Johnson Algorithm

(Total Lead Time for A→B)=a1+b2+max{a2,b1}

(Total Lead Time for B→A)=b1+a2+max{b2,a1}

Assume that total Processing Time for A→B is shorter.

a1+b2+max{a2, b1}≦b1+a2+max{b2, a1}

max{-b1,-a2}≦max{-b2,-a1}

-min{b1,a2}≦-min{b2,a1}

min{b1,a2} ≧ min{b2,a1}

⇔ Processing Time of a1 or b2 is shortest.

According to Johnson Algorithm, all pairs of direct neighboring jobs satisfy the above-mentioned condition

∴ Johnson Algorithm derives the minimum makespan solution.

Scheduling Methods for Production Scheduling Problems

- Method to derive a near optimum solution
  - Meta Heuristic Methods
  - Dispatching Rule Method

    a method which derives a good solution empirically.
- Method to derive an Optimum Solution
  - Jackson Algorithm
  - Branch and Bound Method

    a search method for excluding solution space which is no need to search

Jackson Algorithm

This is applied for JSSP which uses 2 machines for each job.

Step1: In the following manner, jobs are categorized.

A: jobs only using Machine A

B: jobs only using Machine B

AB: jobs using Machine A first and Machine B next

BA: jobs using Machine B first and Machine A next

Step2: Jobs AB and BA are ordered by Johnson Algorithm.

Step3: For each Machine, in the following manner, Jobs are ordered.

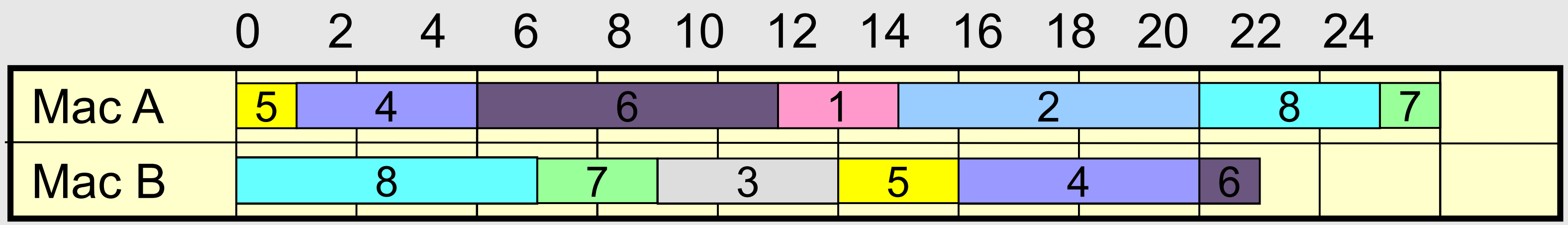
A: AB A BA

B: BA B AB

Scheduling Result using Jackson Algorithm

Type	Job	Ope. Time		
		Mac A		Mac B
A	1	2		—
	2	5		—
B	3	—	—	3
AB	4	3	→	4
	5	1	→	2
	6	5	→	1
BA	7	1	←	2
	8	3	←	5

Machine A Order :
5 4 6 1 2 8 7
Machine B Order :
8 7 3 5 4 6



SMART
INDUSTRY
LABORATORY

Thank you