Chapitre 6

Chapter 6

Ordonnancement d'ateliers

Workshop scheduling

Learning outcomes

- Distinguer la complexité d'un problème d'ordonnancement pour mettre en œuvre une méthode de résolution adaptée *Distinguish the complexity of a scheduling problem to implement an appropriate resolution method*

- Caractériser les problèmes d'ordonnancement en atelier pour sélectionner la méthode la plus appropriée selon le critère à optimiser *Characterize single workshop scheduling problems to select the most appropriate method according to the criterion to be optimized*

- Concevoir des modèles (mathématiques et de simulation) et les exploiter pour évaluer et résoudre des problèmes d'ordonnancement offline avec un solveur ou un simulateur
  Design models (mathematical and simulation) and use them to evaluate and solve offline scheduling problems with a solver or a simulator

- Sélectionner, appliquer et programmer des méthodes d'optimisation pour résoudre les problèmes d'ordonnancement
  Select, apply and program optimization methods to solve scheduling problems

Plan

- Introduction
- FLOW SHOP workshop
  - Mathematical model
  - Cmax

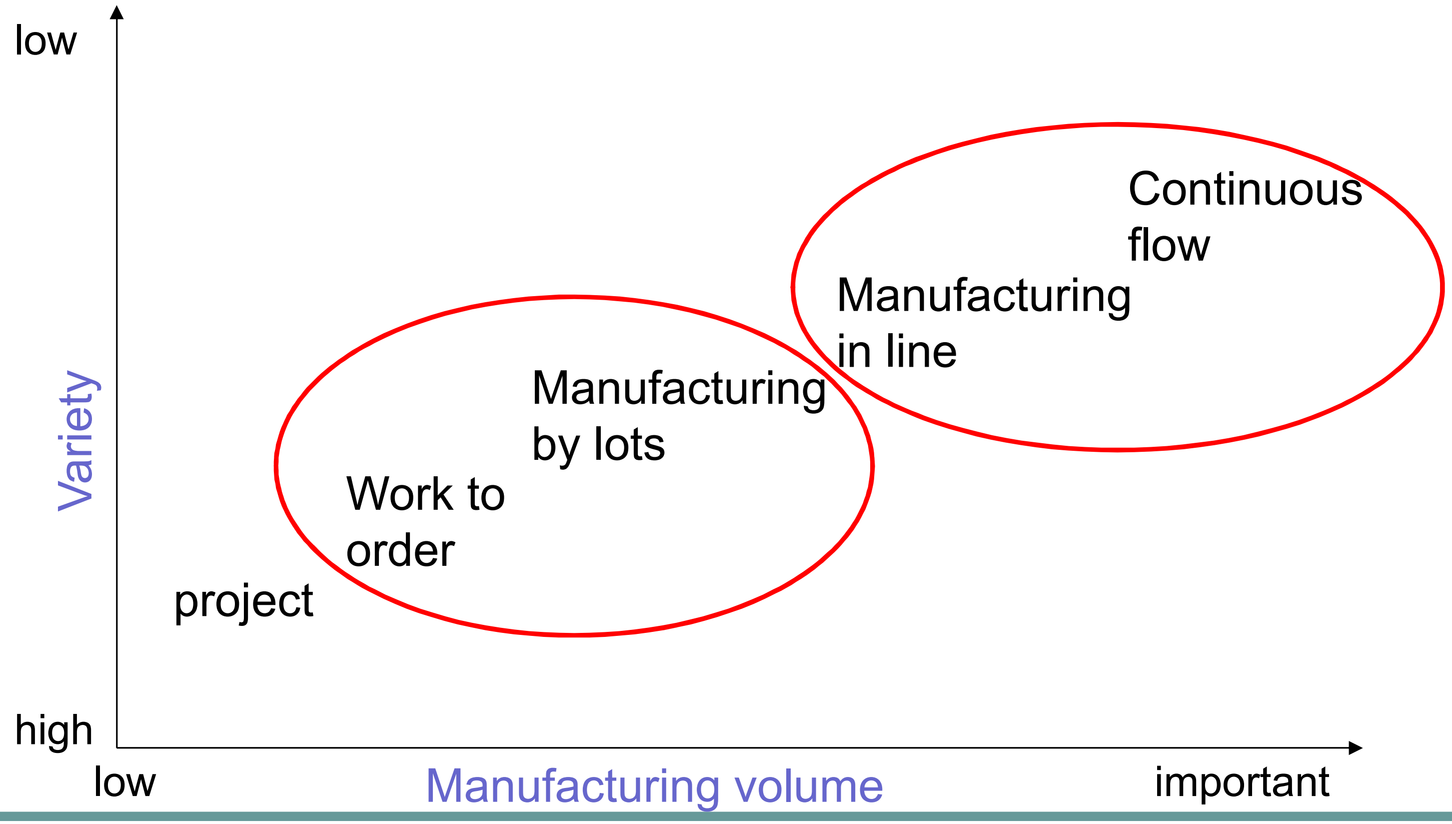
Plan

- Introduction
- FLOW SHOP workshop
  - Mathematical model
  - Cmax

Introduction

- Study of single machine scheduling :
  - Basic problem
  - Study of different criteria
  - Basis for study of more complicated system
- Study of parallel machine scheduling *(coming soon)*
- Study of production line :
  - Different types

Introduction

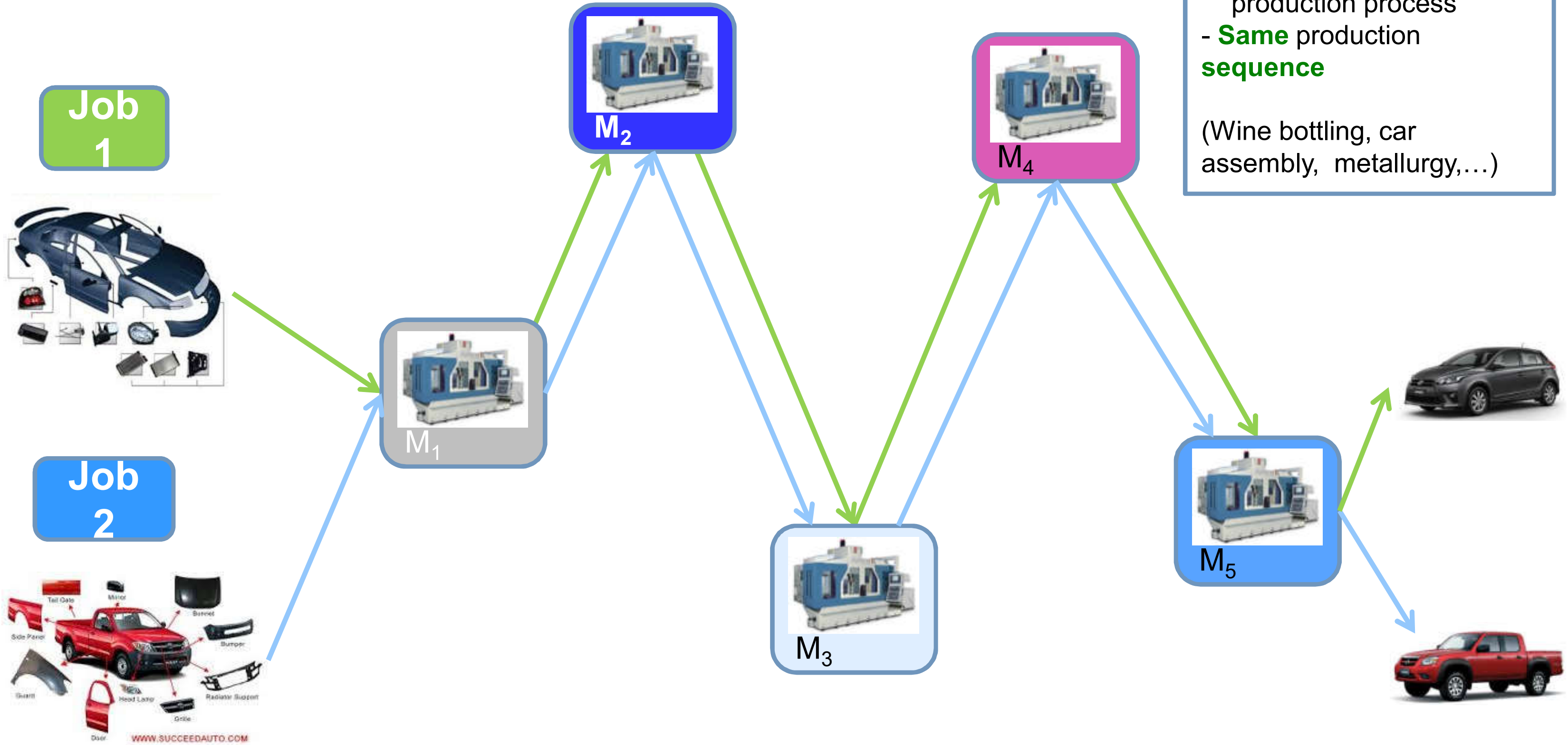


Introduction

- Flow shop
  - Machines are used by the jobs are **in the same order**
- Job shop
  - Machines are **not used in the same order**
- Open shop
  - **No precedence relation** between operations

Introduction

FLOW SHOP



- **High** production **volume**
- **Low variability** in the production process
- **Same** production **sequence**

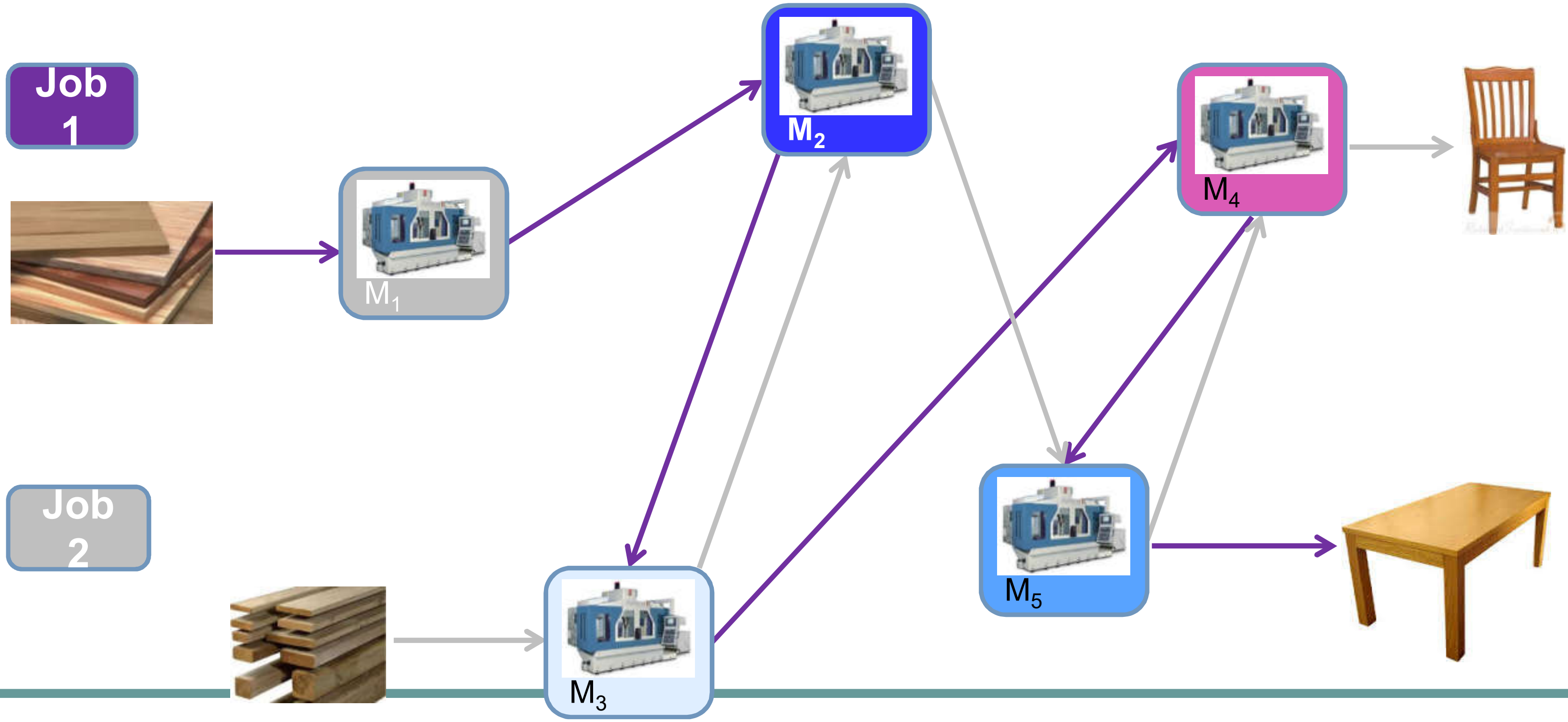
(Wine bottling, car assembly, metallurgy,…)

Introduction

- **Low** production **volume**
- **High variability** in the production process
- **Different** production **sequence** by job

(Carpenter's workshop, hospitals, restaurants,…)

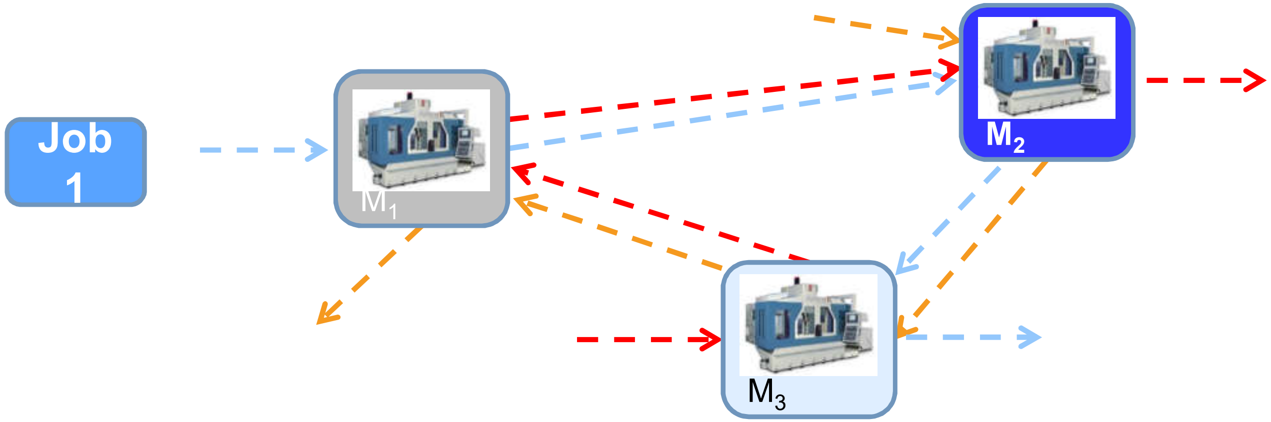
JOB SHOP



Introduction

OPEN SHOP :

- **Multiple possible** production **sequences** by job
- The production sequence is **immaterial**



Introduction

- Diversities of the scheduling problems
  - Machines nature
  - Manufacturing technics
  - Production workshop
  - Criteria
  - Nature of requirement (dynamic or constant, determinist or stochastic)
  - Machines characteristics

Plan

- Introduction
- FLOW SHOP workshop
  - Mathematical model
  - Cmax

Modélisation

- Permutation scheduling :
  - 2 products p_1 and p_2 are processed in a Flow shop ; if p_1 *is* before p_2 on M1, that implies the same order on all the other machines

- N products and M machines :
  - If permutation scheduling : $N!$ possibilities,
  - If not permutation scheduling : $(N!)^M$ possibilities

À vos stylos!

Plan

- Introduction
- FLOW SHOP workshop
  - Mathematical model
  - Cmax

Flow Shop, C_{max}

- Particular case M=2
  - Optimal solution = permutation schedule following

    Johnson rule :
    - job i precedes job j in the optimal sequence of the flow

      shop with 2 machines if :

$$\boldsymbol{min\{p_{i,1}, p_{j,2}\} < min\{p_{i,2}, p_{j,1}\}}$$

Flow Shop, C_{max}

- Application of Johnson rule (1954) :
  - Put the jobs into two lists : S_{deb} and S_{fin}
    - The jobs i with $p_{i,1} < p_{i,2}$ are in S_{deb}
    - The jobs i with $p_{i,1} > p_{i,2}$ are in S_{fin}
  - The jobs in S_{deb} are scheduled in the increasing of $p_{i,1}$
  - The jobs in S_{fin} are scheduled in the decreasing order of $p_{i,2}$
  - The jobs in S_{deb} are processed before those in S_{fin}

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
$p_{i,1}$	5	1	9	3	10
$p_{i,2}$	2	6	7	8	4

- S_{deb} ={2,4}
- S_{fin} ={1,3,5}
- S_{deb} =2 – 4
- S_{fin} =3 – 5 – 1
- Optimale solution = 2 – 4 – 3 – 5 – 1

Flow Shop, C_{max}

- Particular case M=3
  - NP-hard
  - We can prove that only the permutation schedules should be considered
  - Two particular cases : optimal solution in polynomial time

$$\min_k \{p_{k,1}\} \geq \max_k \{p_{k,2}\}$$

time on M1 more than on M2

$$\min_k \{p_{k,3}\} \geq \max_k \{p_{k,2}\}$$

time on M3 more than on M2

Machine M2 in the middle not dominant (not critical)

Flow Shop, C_{max}

- So *i* before *j* in the optimal sequence if

$$\min\{p_{i,1} + p_{i,2}, p_{j,2} + p_{j,3}\} \leq \min\{p_{i,2} + p_{i,3}, p_{j,1} + p_{j,2}\}$$

Flow Shop, C_{max}

- General case with m machines
  - Examples of approximated solutions :
    - **Palmer** heuristic
    - **Gupta** heuristic
    - **CDS** (Campbell, Dudek, Smith) Heuristic
    - **NEH** (Nawaz, Enscore, Ham) heuristic

Flow Shop, C_{max}

- Palmer heuristic

  - For each job we compute :

$$s_j = \sum_{k=1}^{m} (2k - m - 1) p_{j,k}$$

  - Schedule the jobs on the machines in the decreasing order of s_j

Flow Shop, C_{max}

- Palmer heuristic
  - $s_i = \sum_{k=1}^{m}(2k - m - 1)p_{ik}$

$$= (2m - m - 1)p_{im} + (2(m-1) - m - 1)p_{im_{-1}} + \ \ldots \ + (2 \times 2 - m - 1)p_{i2} + \ (2 \times 1 - m - 1)p_{i1}$$

$$= (m-1)p_{im} + (m-3)p_{im_{-1}} + \cdots + (3-m)p_{i2} + \qquad (1-m)p_{i1}$$

$$= (m-1)p_{im} + (m-3)p_{im_{-1}} + \cdots - (m-3)p_{i2} - (m-1)p_{i1}$$

Flow Shop, C_{max}

- Palmer heuristic
  - $s_i = \sum_{k=1}^{m}(2k - m - 1)p_{ik}$

    $$= (m-1)p_{im} + (m-3)p_{im_{-1}} + \cdots - (m-3)p_{i2} - (m-1)p_{i1}$$
  - For the same duration in the end of process :
    - Shorter jobs at the beginning → higher score
  - For the same duration in the beginning of process :
    - Longer jobs at the end → higher score

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i					

$$s_i = \sum_{k=1}^{m} (2k - m - 1) p_{ik} = 2p_{i3} - 2p_{i1}$$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i					

$= 2 \times 4 - 2 \times 5 = -2$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-2				

$= 2 \times 2 - 2 \times 1 = 2$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-2	2			

$= 2 \times 3 - 2 \times 9 = -12$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-2	2	-12		

$= 2 \times 5 - 2 \times 3$ =4

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-2	2	-12	4	

$= 2 \times 3 - 2 \times 10 = -14$

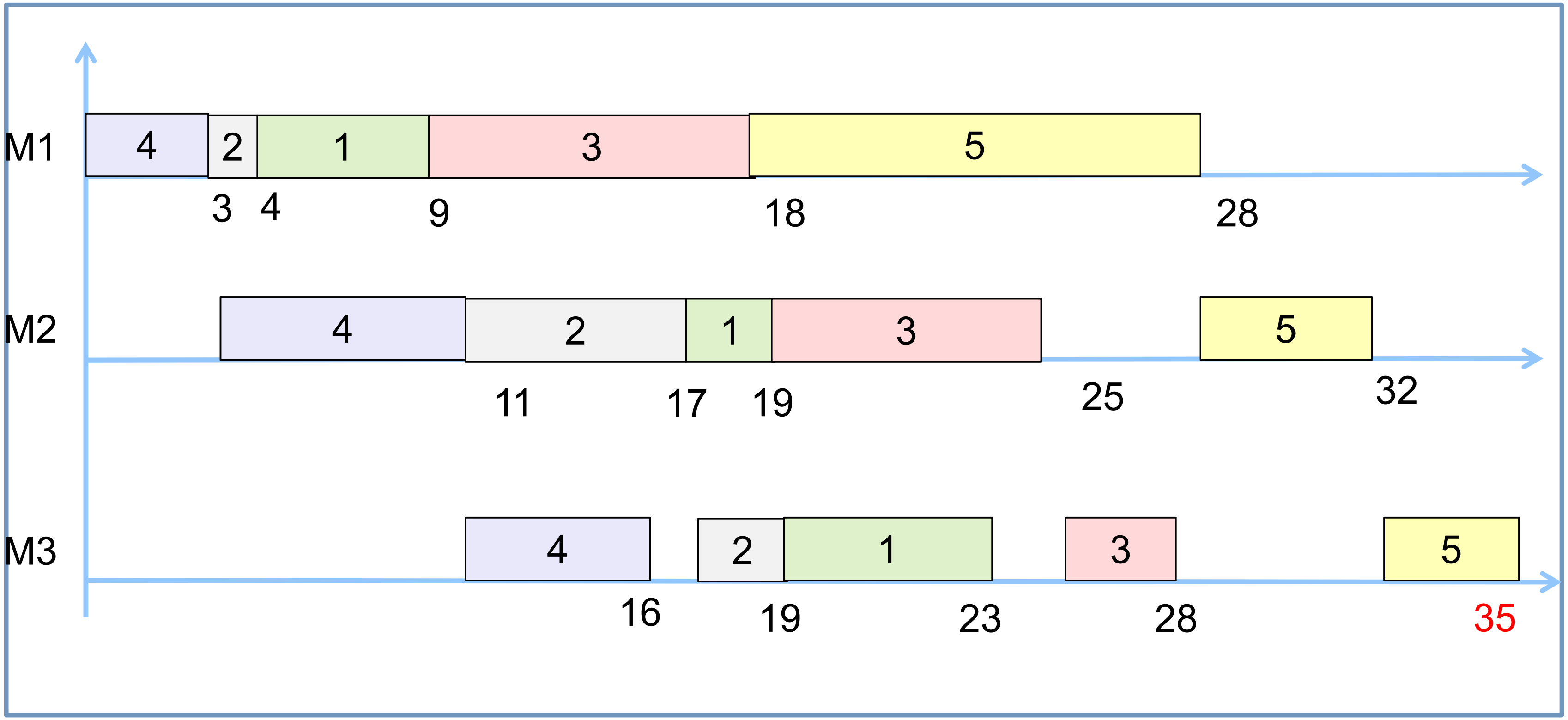
Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-2	2	-12	4	-14

- No-increasing order
- 4 - 2 - 1 - 3 - 5

Flow Shop, C_{max}



Flow Shop, C_{max}

- **Gupta heuristic**

$$s_j = \frac{e_j}{\min\limits_{1 \le k \le m-1} \{ p_{j,k} + p_{j,k+1} \}}$$

- With $e_j = 1$ if $p_{j,1} < p_{j,m}$, -1 otherwise
- Decreasing order of s_j :
  - Priority for shorter jobs on the 1st machine
  - Priority for shorter jobs on the intermediate machines

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i					

$$s_i = \frac{e_i}{\min\limits_{1 \le k \le m-1} \{p_{ik} + p_{ik+1}\}} = \frac{-1}{6} \cong -0{,}16$$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-0,16				

$$s_i = \frac{e_i}{\min_{1 \le k \le m-1} \{p_{ik} + p_{ik+1}\}} = \frac{1}{7} \cong 0{,}14$$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-0,16	0,14			

$$s_i = \frac{e_i}{\min\limits_{1\leq k\leq m-1}\{p_{ik}+p_{ik+1}\}} = \frac{-1}{10} \cong -0{,}1$$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-0,16	0,14	-0,1		

$$s_i = \frac{e_i}{\min\limits_{1 \le k \le m-1} \{p_{ik} + p_{ik+1}\}} = \frac{1}{11} \cong 0{,}09$$

Flow Shop, C_{max}

- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-0,16	0,14	-0,1	0,09	

$$s_i = \frac{e_i}{\min\limits_{1\le k\le m-1}\{p_{ik}+p_{ik+1}\}} = \frac{-1}{7} \cong -0{,}14$$

Flow Shop, C_{max}

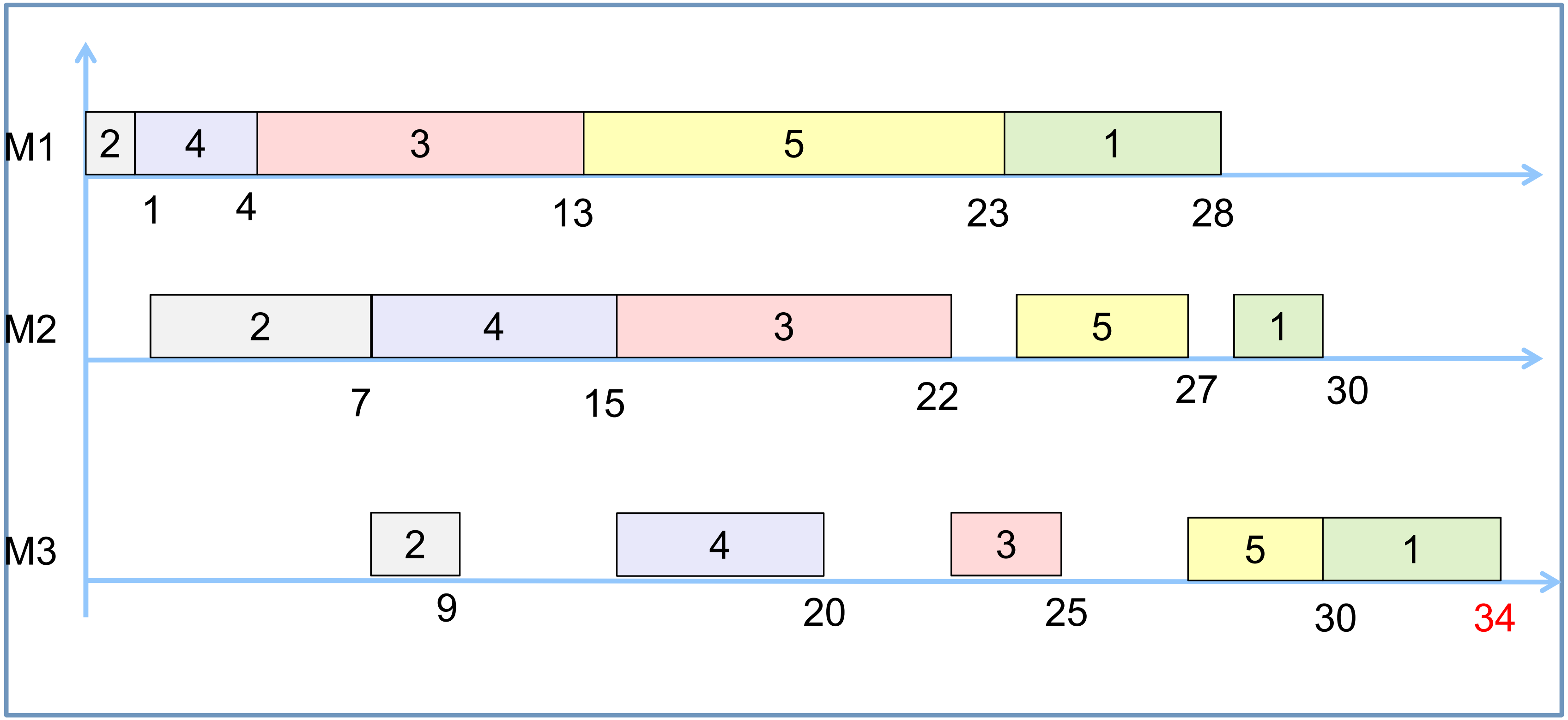
- Example

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
s_i	-0,16	0,14	-0,1	0,09	-0,14

- Decreasing order : 2-4-3-5-1

Palmer : 4 - 2 - 1 - 3 - 5

Flow Shop, C_{max}



Flow Shop, C_{max}

- **CDS heuristic**
  - For *j* from 1 to m-1 do
    - Consider the problem with 2 fictive machines (M1, M2) where
      - duration on M1: sum of durations on the machines 1 to *j*
      - duration on M2: sum of durations on the machines *m*+1-*j* to *m*
    - Apply Johnson rule to M1 and M2,
    - Deduce the jobs schedule
    - Compute the total duration for the initial problem
  - End for
  - Take the solution with the best criterion value

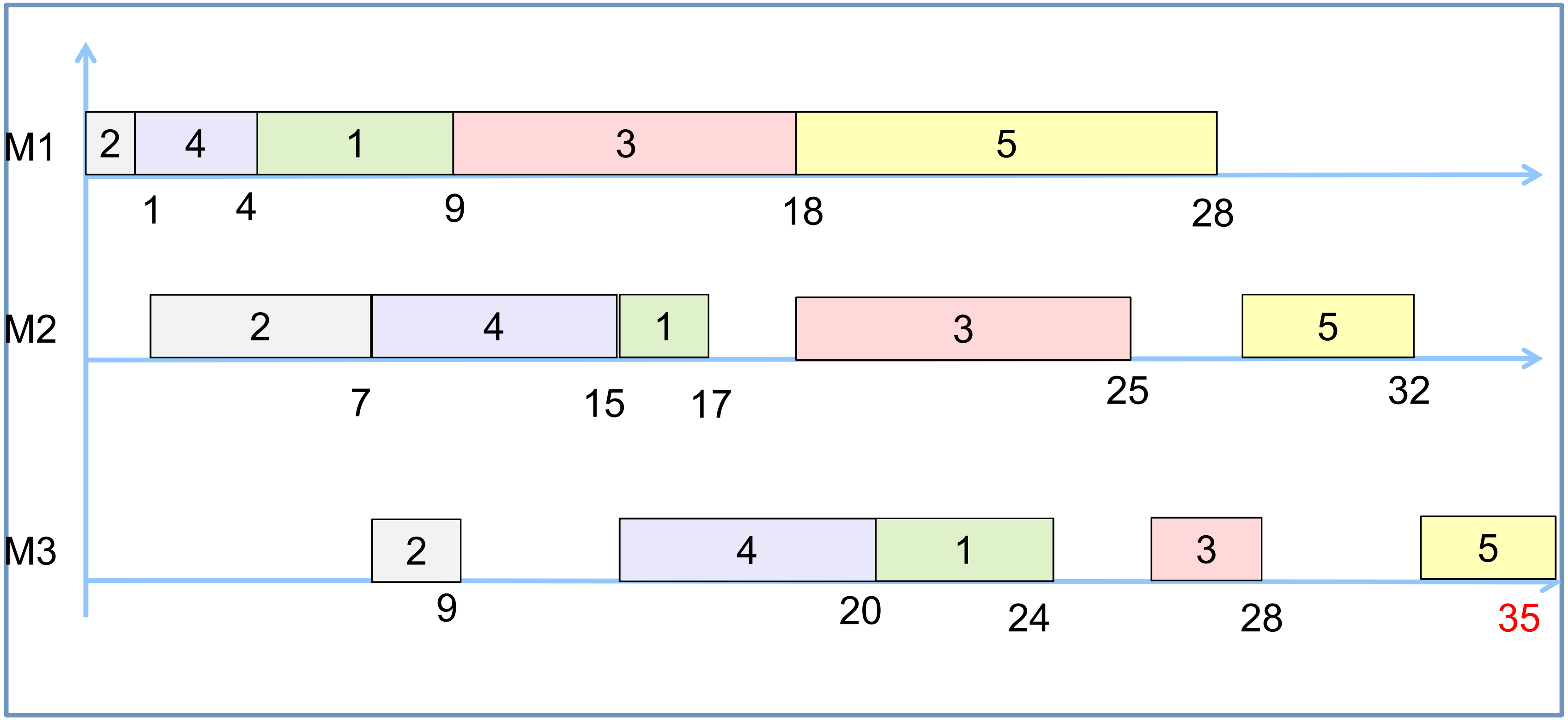
Flow Shop, C_{max}

- Example : j=1

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
$M_1(1 \rightarrow 1)$	5	1	9	3	10
$M_2(3 \rightarrow 3)$	4	2	3	5	3
	SF	SD	SF	SD	SF

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

Flow Shop, C_{max}



Flow Shop, C_{max}

- Example : j=2

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
$M_1(1 \to 2)$	7	7	16	11	14
$M_2(2 \to 3)$	6	8	10	13	7
	SF	SD	SF	SD	SF

- Order : 2-4-3-5-1 the same as Gupta, Cmax=34
- Keep this order, better as the one obtained with j=1

Flow Shop, C_{max}

- **NEH heuristic**
  - $\forall j = 1, \dots n \qquad T_j = \sum_{i=1}^{m} p_{ji}$
  - $\rho = \left(\rho(1), \rho(2), \dots \rho(K)\right)$: a partial schedule where $K = card(\rho)$
  - $\rho(j,k)$: partial schedule constituted by ρ and the j inserted in the position k:
    - Examples :

$$\rho(j,1) = (j,\rho)$$

$$\rho(j,k) = (\rho(1), \rho(2), \dots, \rho(k-1), j, \rho(k) \dots \rho(K))$$

$$\rho(j,k+1) = (\rho, j)$$

Flow Shop, C_{max}

- **Step 1**:
  - Sort the jobs in the decreasing order of T_j to find a priority order $\gamma = (\gamma(1), \gamma(2), \dots \gamma(n))$
  - $\rho = \gamma(1)$
  - $K = 1$
- **Step 2**:
  - $j = \gamma(K+1)$
  - For $k = 1, \dots K+1$
    - Compute $C_{max}(\rho(j,k))$
  - Keep the position k^* associated to $\rho(j,k^*)$ that minimizes $C_{max}(\rho(j,k))$
  - In case of equality, k^* is the first index came across

Flow Shop, C_{max}

- **Step 3**:
  - $\rho = \rho(j, k^*)$ and $K = K + 1$
  - If $K < n$, return to step 2
  - Otherwise ρ is the final sequence.

- Performances of the method ?
  - One of the best heuristics (**7% of the optimum**)
  - Improvements (**NEH KK1**, **NEH KK2**)

Flow Shop, C_{max}

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
T_i	11	9	19	16	17

- Example
- Order : 3-5-4-1-2
- We place job 3 :

$(9 \;\; 16 \;\; 19)$

- Job 5 insertion :
  - Ordre 5-3 : $\begin{pmatrix} 10 & 14 & 17 \\ 19 & 26 & 29 \end{pmatrix}$
  - Ordre 3-5 : $\begin{pmatrix} 9 & 16 & 19 \\ 19 & 23 & 26 \end{pmatrix}$

Flow Shop, C_{max}

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
T_i	11	9	19	16	17

- Order 3-5 : $\begin{pmatrix} 9 & 16 & 19 \\ 19 & 23 & 26 \end{pmatrix}$
- Job 4 insertion :

4-3-5 : $\begin{pmatrix} 3 & 11 & 16 \\ 12 & 19 & 22 \\ 22 & 26 & 29 \end{pmatrix}$

3-4-5: $\begin{pmatrix} 9 & 16 & 19 \\ 12 & 24 & 29 \\ 22 & 28 & 32 \end{pmatrix}$

3-5-4 : $\begin{pmatrix} 9 & 16 & 19 \\ 19 & 23 & 26 \\ 22 & 31 & 36 \end{pmatrix}$

Flow Shop, C_{max}

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
T_i	11	9	19	16	17

- 4-3-5 : $\begin{pmatrix} 3 & 11 & 16 \\ 12 & 19 & 22 \\ 22 & 26 & 29 \end{pmatrix}$
- Job 1 insertion :
- 1-4-3-5 $\begin{pmatrix} 5 & 7 & 11 \\ 8 & 16 & 21 \\ 17 & 24 & 27 \\ 27 & 31 & 34 \end{pmatrix}$ 4-1-3-5 $\begin{pmatrix} 3 & 11 & 16 \\ 8 & 13 & 20 \\ 17 & 24 & 27 \\ 27 & 31 & 34 \end{pmatrix}$
- 4-3-1-5 $\begin{pmatrix} 3 & 11 & 16 \\ 12 & 20 & 25 \\ 17 & 22 & 29 \\ 27 & 31 & 34 \end{pmatrix}$ 4-3-5-1 $\begin{pmatrix} 3 & 11 & 16 \\ 12 & 20 & 25 \\ 22 & 26 & 29 \\ 27 & 29 & 33 \end{pmatrix}$

Flow Shop, C_{max}

i	1	2	3	4	5
p_{i1}	5	1	9	3	10
p_{i2}	2	6	7	8	4
p_{i3}	4	2	3	5	3
T_i	11	9	19	16	17

- 4-3-5-1 $\begin{pmatrix} 3 & 11 & 16 \\ 12 & 20 & 25 \\ 22 & 26 & 29 \\ 27 & 29 & 33 \end{pmatrix}$
- Job 2 insertion :

2-4-3-5-1 : C_{max} =34, 4-2-3-5-1 $\begin{pmatrix} 3 & 11 & 16 \\ 4 & 17 & 19 \\ 13 & 24 & 27 \\ 23 & 28 & 31 \\ 28 & 30 & 35 \end{pmatrix}$ 4−3−2−5−1 $\begin{pmatrix} 3 & 11 & 16 \\ 12 & 20 & 25 \\ 13 & 26 & 28 \\ 23 & 30 & 33 \\ 24 & 32 & 37 \end{pmatrix}$

4−3−5−2−1 $\begin{pmatrix} 3 & 11 & 15 \\ 12 & 20 & 25 \\ 22 & 26 & 29 \\ 23 & 32 & 34 \\ 28 & 34 & 38 \end{pmatrix}$ 4−3−5−1−2 $\begin{pmatrix} 3 & 11 & 16 \\ 12 & 20 & 25 \\ 22 & 26 & 29 \\ 27 & 29 & 33 \\ 28 & 35 & 37 \end{pmatrix}$