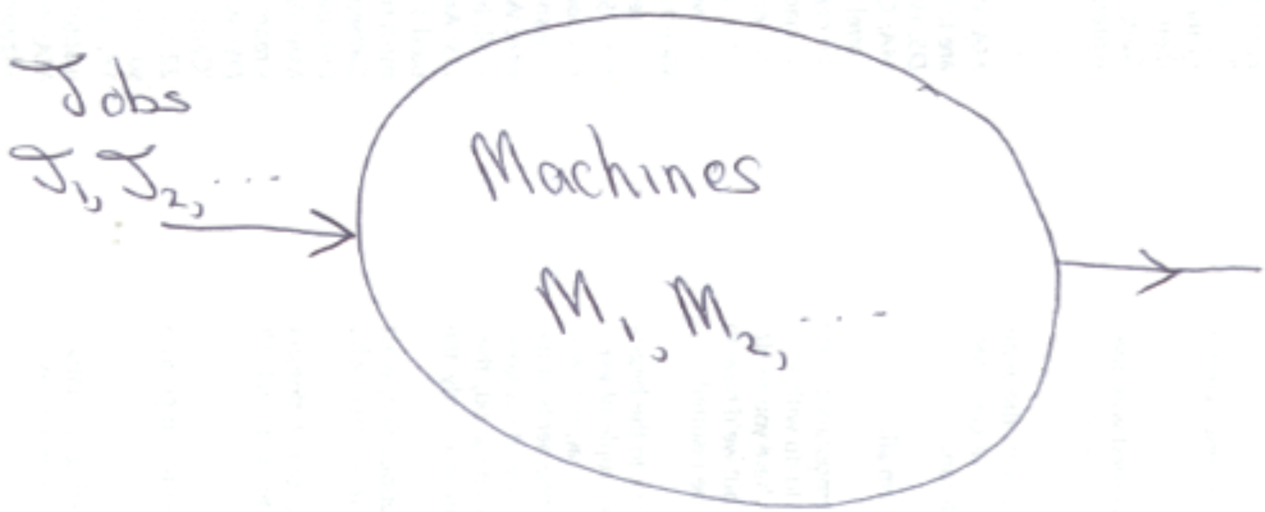
Job Shop Scheduling

Machine Shop



Example 1

$$1 \mid \cdot\cdot \mid \sum_j w_j C_j$$

#of jobs restrictions objective

C_j = completion time of job j

p_j = processing time of job j

What is the best ordering of the jobs to minimise the objective?

$$\boxed{n} \quad \cdots \quad \boxed{2} \quad \boxed{1} \quad \boxed{M}$$

$$\frac{w_1}{p_1} \geqslant \frac{w_2}{p_2} \geqslant \cdots \geqslant \frac{w_n}{p_n}$$

Suppose we do not use this order.

Only j, k get new completion times

$$\text{New} - \text{Old} = w_k(t+p_k) + w_j(t+p_k+p_j)$$
$$- w_j(t+p_j) + w_k(t+p_j+p_k)$$
$$= w_j p_j - w_k p_j < 0.$$

Example 2

$1 \mid \cdot\cdot \mid L_{max}$

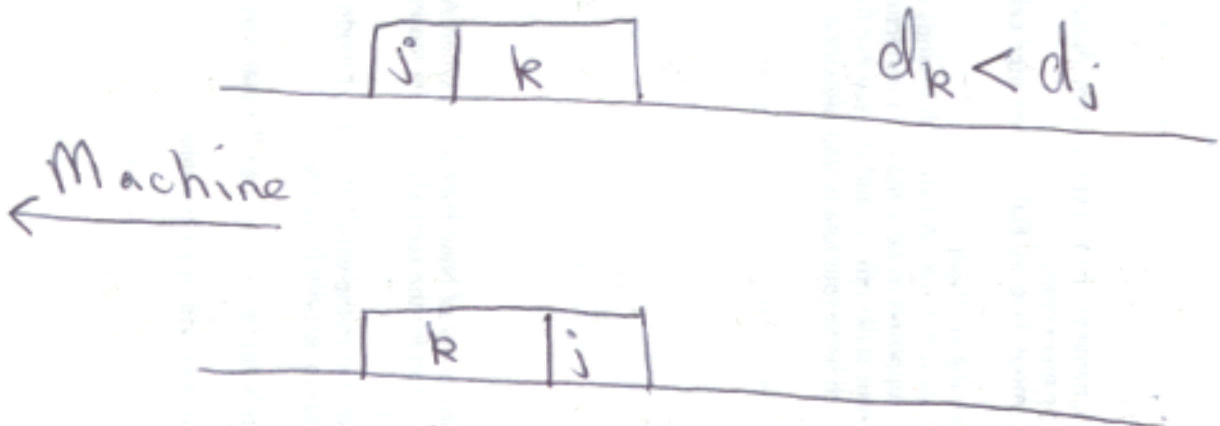
Job j has due date d_j

$L_j = (C_j - d_j)^+$

$L_{max} = \max_j L_j$

Sort so that $d_1 \leq d_2 \leq \cdots \leq d_n$

Suppose we do not use this order.



$d_k < d_j$

Contribution of j and k

Old: $\max\left((C_j^{old} - d_j)^+, (C_k^{old} - d_k)^+ \right) = (C_k^{old} - d_k)$

New: $\max\left((C_j^{new} - d_j)^+, (C_k^{new} - d_k)^+ \right) \leq$

$C_j^{new} \leq C_k^{old}$, $\quad d_j > d_k$, $\quad C_k^{new} \leq C_k^{old}$

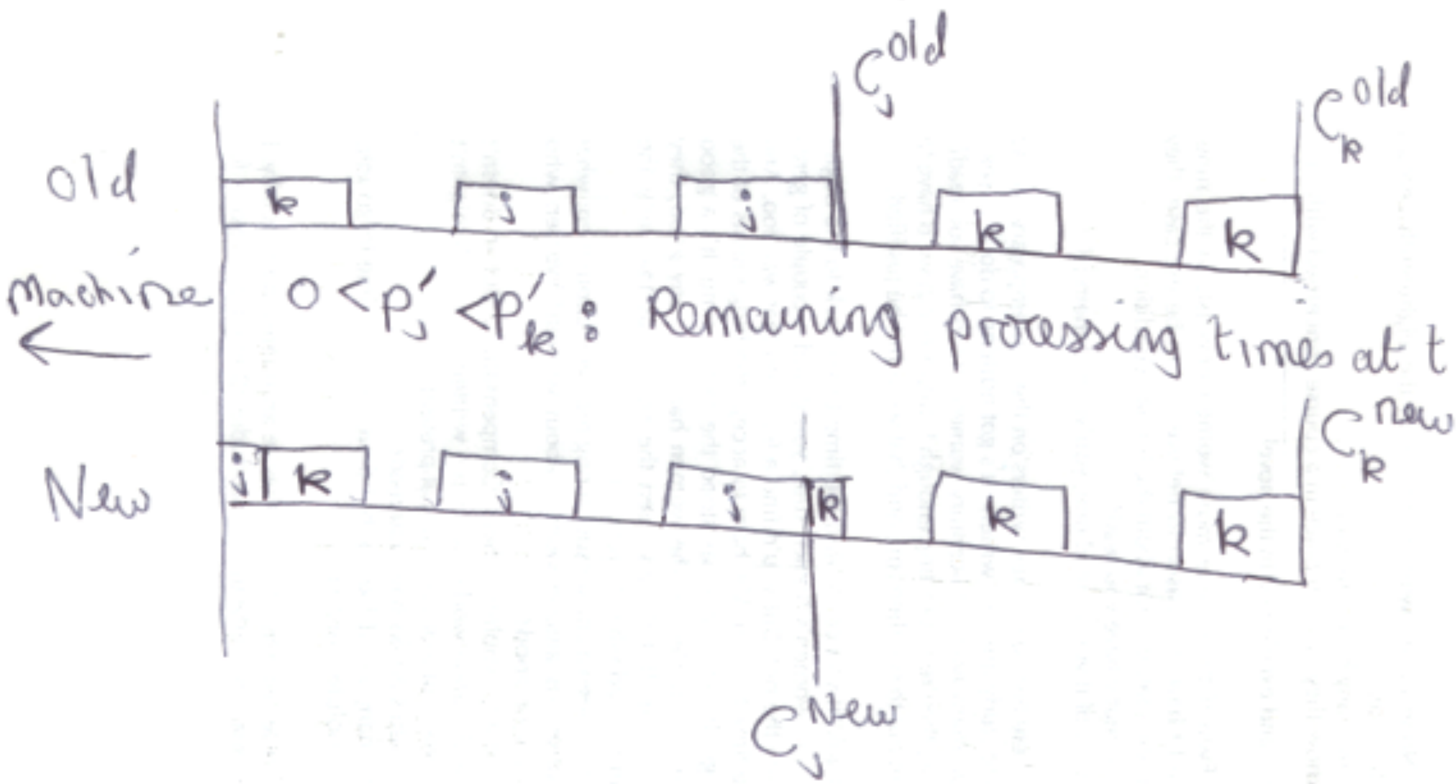
Example 3

$$1 \,|\, r_j,\ \text{premption} \,|\, \sum_j C_j$$

release date

SRPT Rule

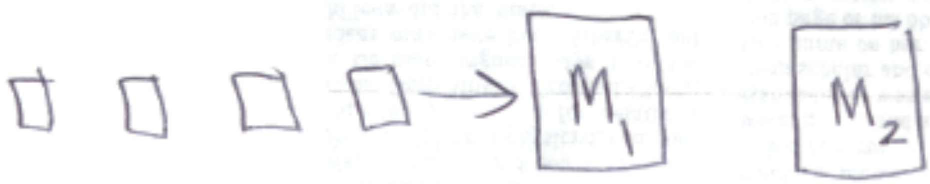
Shortest Remaining Processing Time



Example 4

Two Machine Flow Shop -
Johnson's Rule

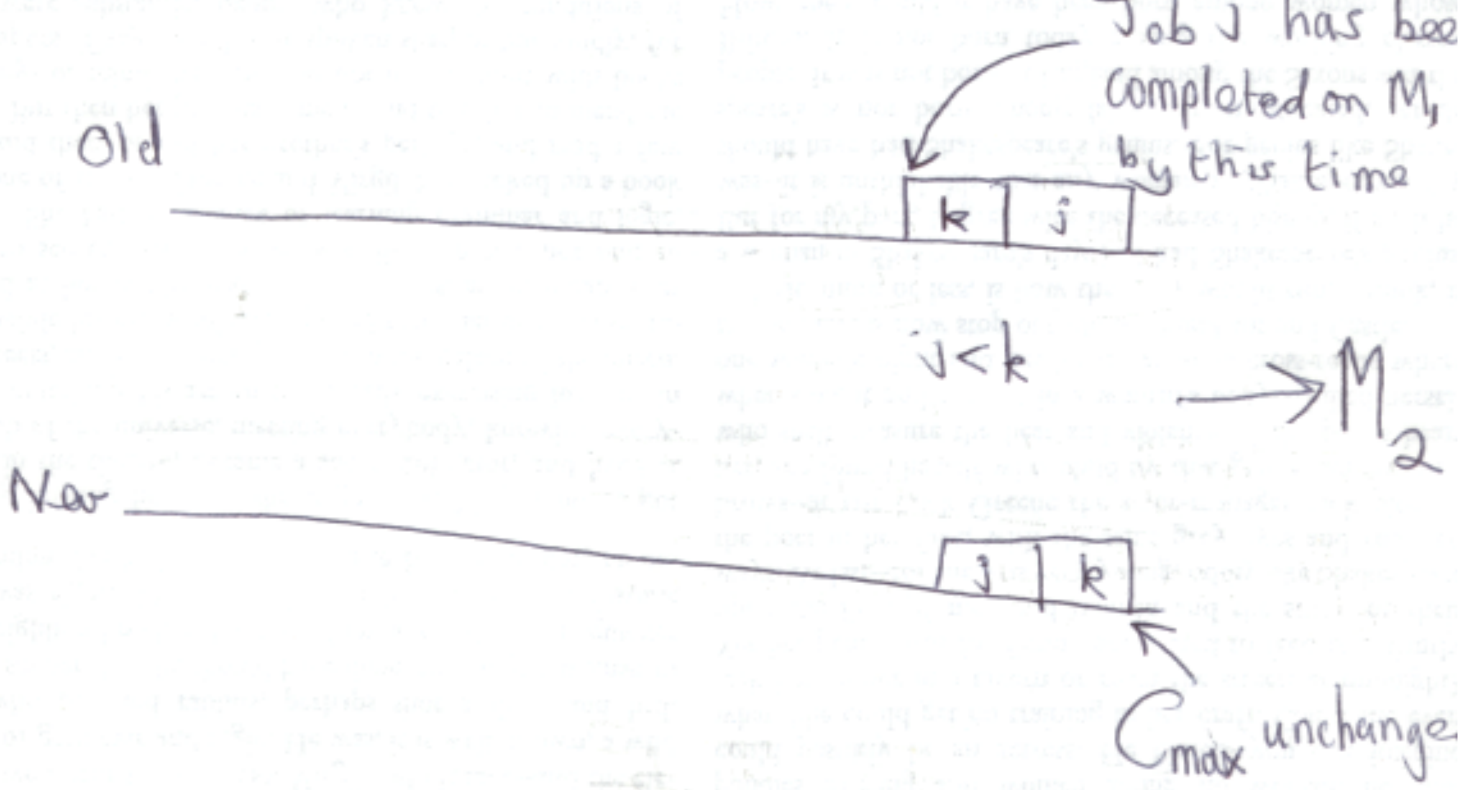
$F2 \mid \ldots \mid C_{max}$



Flow Shop: every job is processed first on M_1, then M_2.

Permutation Flow Shop: Same order on each machine

We can assume a permutation schedule



Processing times: $a_1, a_2, \dots, a_n$ on M_1

$b_1, b_2, \dots, b_n$ on M_2

$$A = \{ j : a_j \leq b_j \}$$

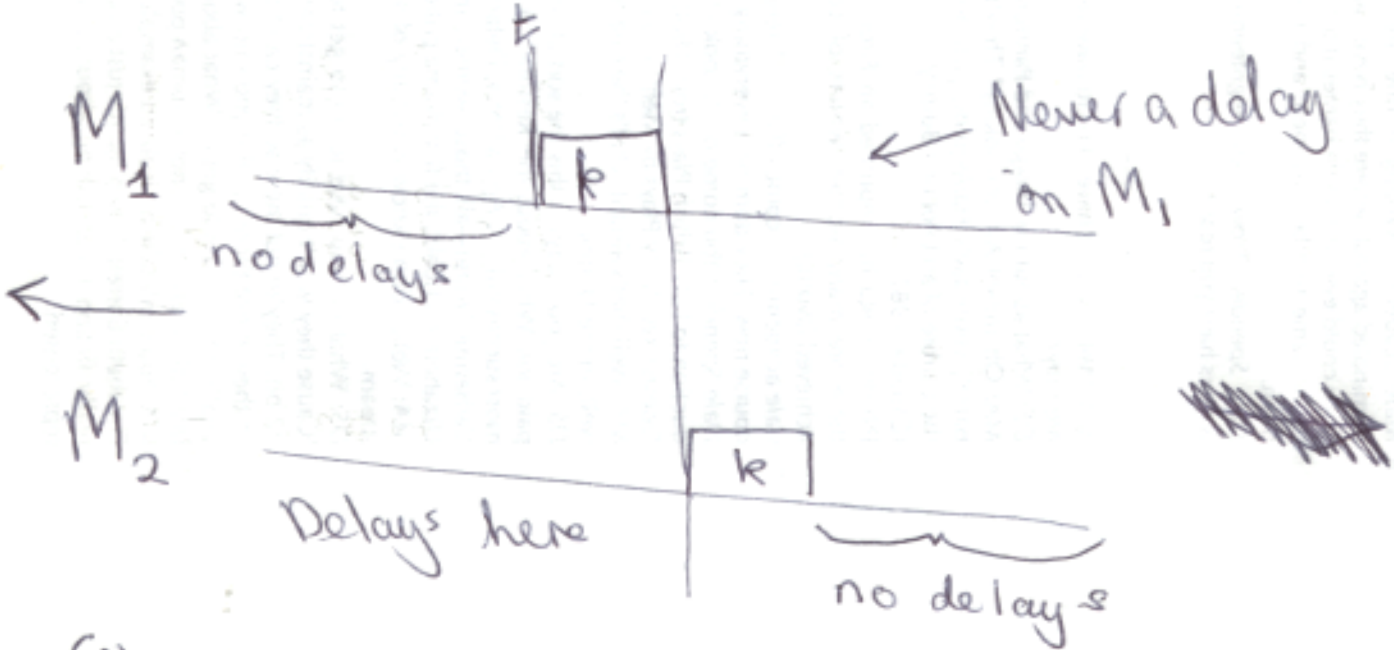
$$B = [n] \setminus A$$

Re-number

$$B \rightarrow A \rightarrow \boxed{M_1}$$

$$b_n \leq b_{n-1} \leq \cdots \leq b_{m+1} \quad a_m \geq a_{m-1} \geq \cdots \geq a_1$$

For all permutation schedules there is a time t

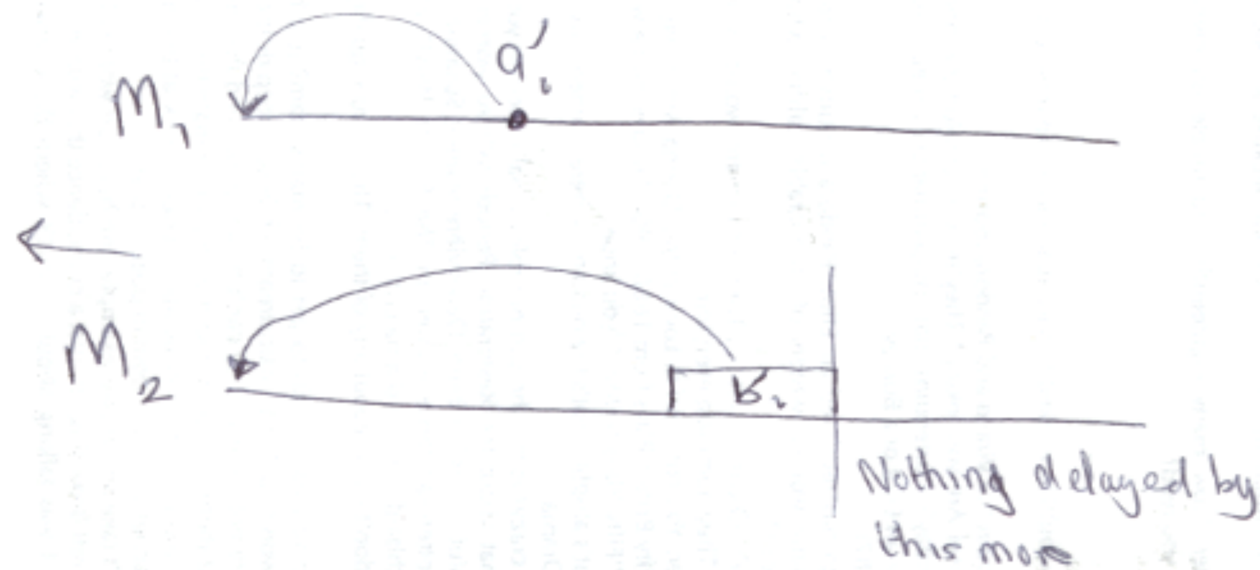


(i) C_{max} = sum of $n+1$ job times.

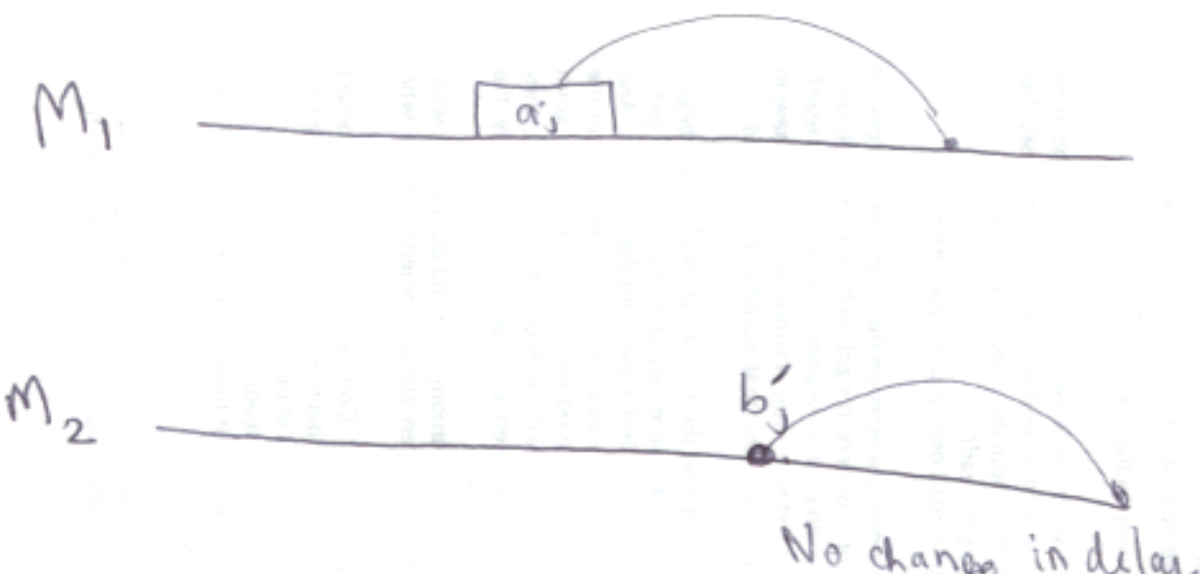
(ii) Reducing every a_i, b_i by same amount ρ does not alter optimal order: — every schedule reduced by $(n+1)\rho$

(iii) $\rho = \min\{a_1, \dots, a_n, b_1, \dots, b_n\}$

(a) $\rho = a_t$: $a_i' = a_i - \rho$, $b_i' = b_i - \rho$. $a_i' = 0$.



(b) $\rho = b_j$: . . . $b'_j = 0$



Above yields Johnson's Rule - induction.