10/26/15

Ex 2

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

Job $j$ has a due date $d_j$

$L_j = (C_j - d_j)^+$

$L_{max} = \max_j L_j$

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

Suppose we do not use this order

Old


$d_k < d_j$

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

& $\max(C_j^{new} - d_j)^+, (C_k^{new} - d_k)^+)$

$C_k^{old} = C_j^{new}$

New

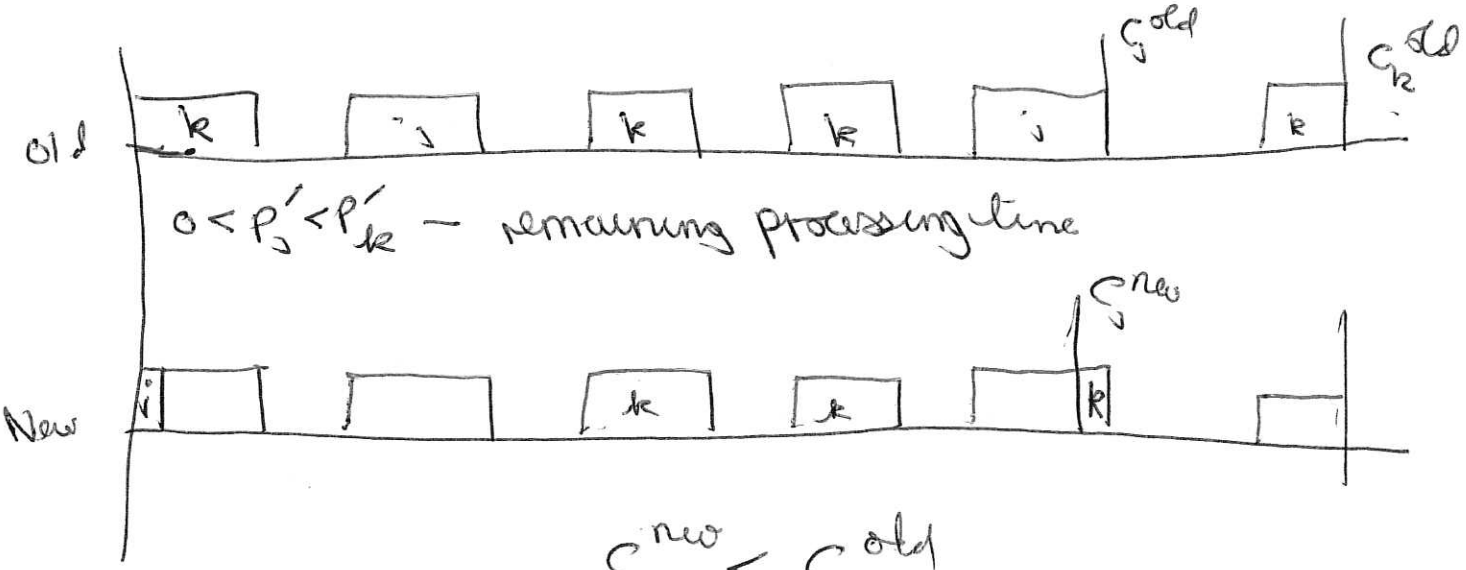
Ex3

$1 \mid r_j, \text{preemption} \mid \sum_j C_j$

$r_j$: release date

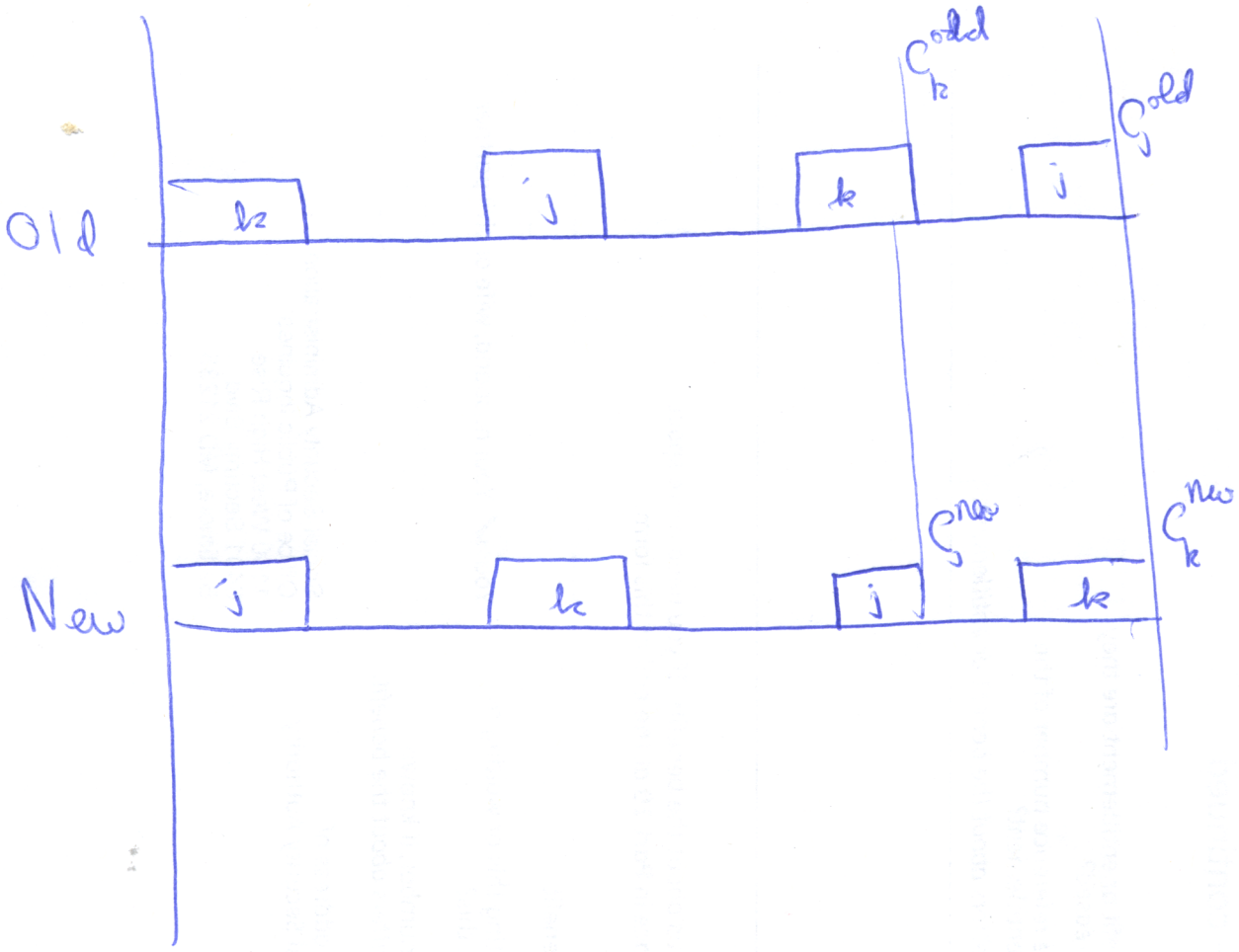
A new job can eject an old job from the machine

Algorithm SRPT rule – shortest remaining processing time

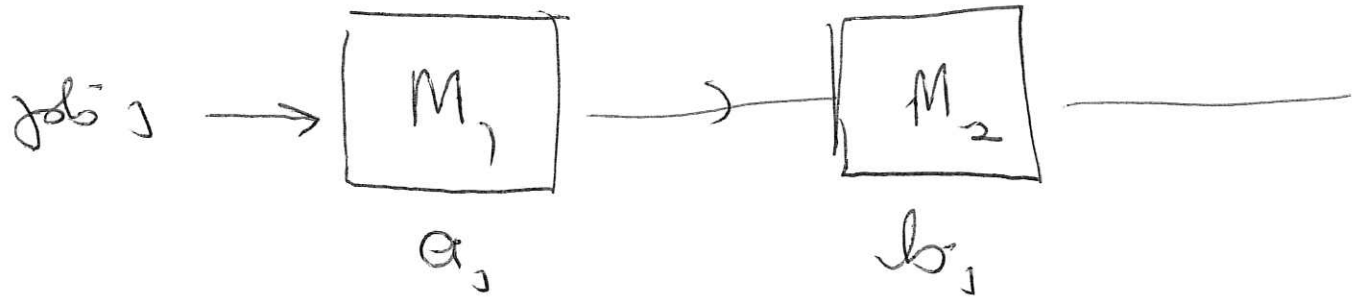


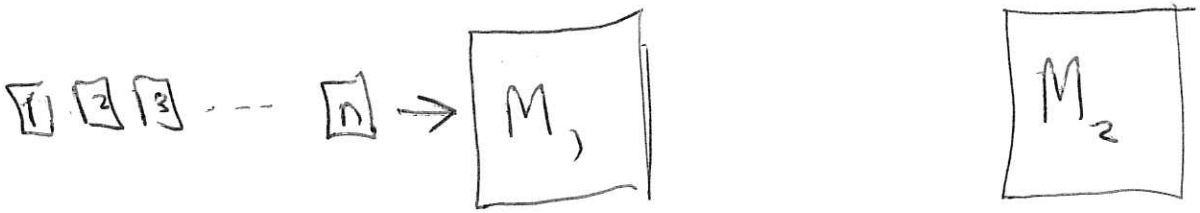

$C_k^{new} < C_j^{old}$

& $C_k^{new} = C_k^{old}$


Old
New
k
j
k
j
j
k
j
k
$C_k^{old}$
$C_j^{old}$
$C_j^{new}$
$C_k^{new}$

# Two machine problem $F2||C_{max}$

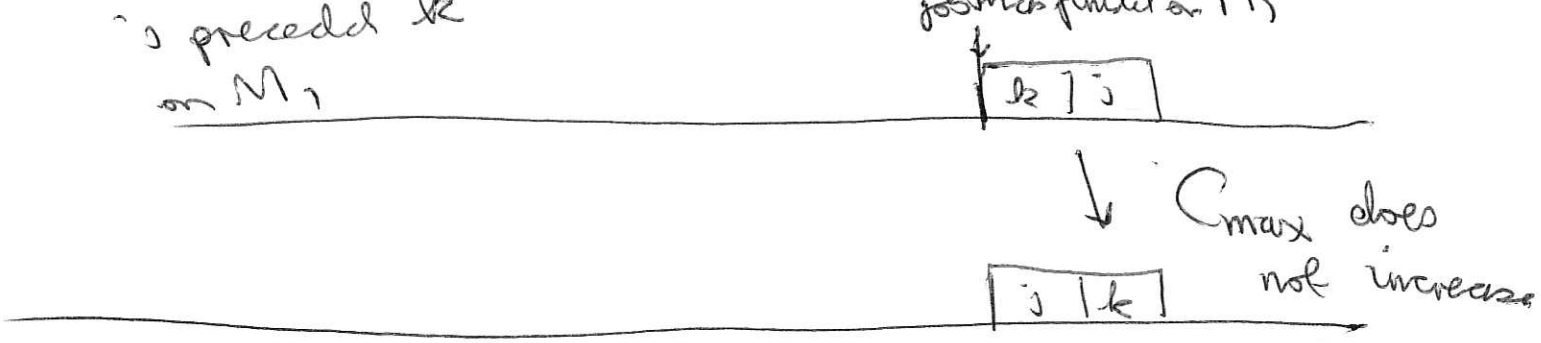



Flow shop: every job is processed by $M_1$ and then $M_2$.

Permutation Flow Shop: same order on each machine

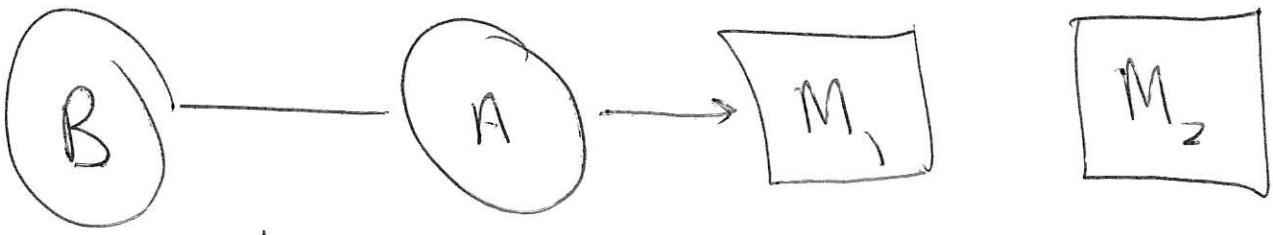
We can assume a permutation flow shop.



# Johnson's Rule

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

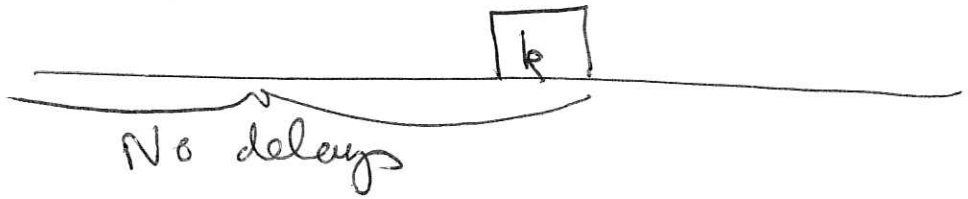
$$B = [n] \setminus A$$



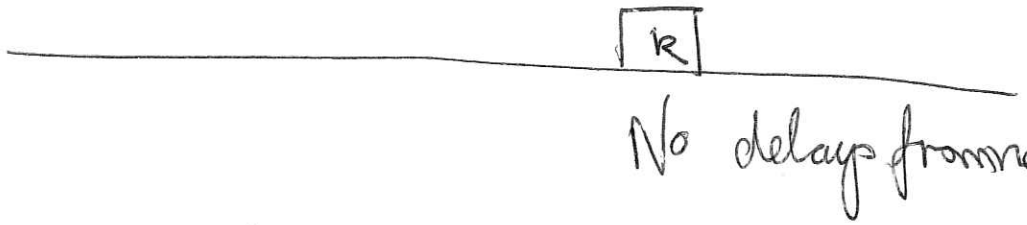

$b_n \leq b_{n-1} \leq \cdots \leq b_{n+1}$ $\quad a_1 \leq a_2 \leq \cdots a_m$

For all permutation schedules:

$M_1$

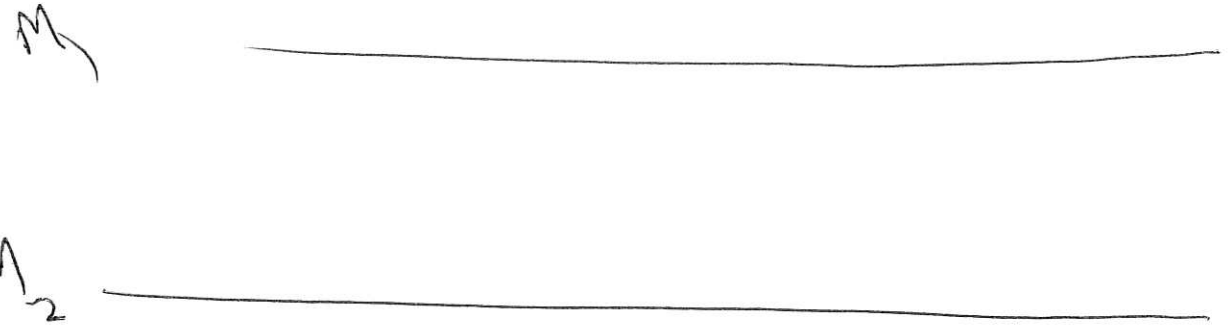



$M_2$



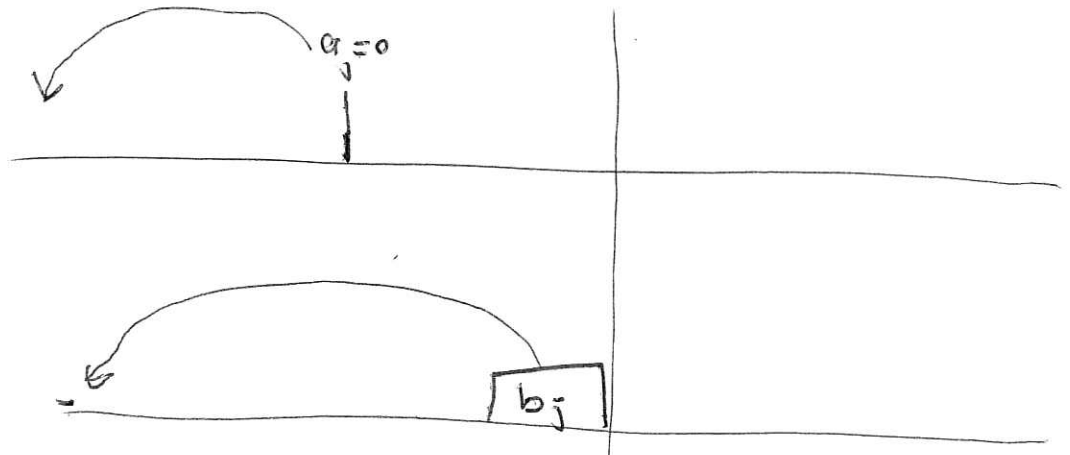

$C_{max}$ = Sum of $n+1$ job time. If you reduce each time by $\lambda$, optimal order doesn't change.

$a_j$ = processing time on $M_1$
$b_j$ = processing time on $M_2$




Reduce by $\lambda$ until say $a_j = 0$.

Claim: put $j$ first.




$b_j = 0$, put it last.