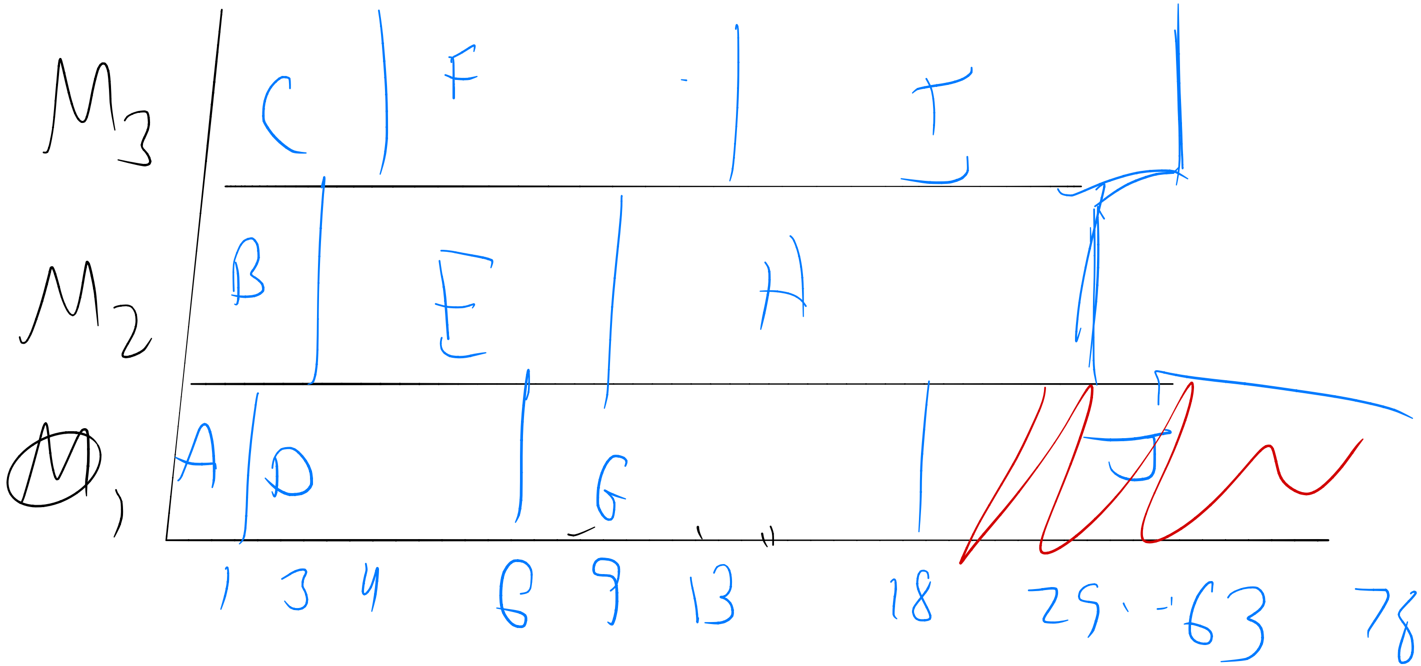
Average Completion Time on Multiple Machines

$P||\sum C_j$

j	p_j
A	**1**
B	**3**
C	**4**
D	**5**
E	**6**
F	**9**
G	**12**
H	**20**
I	**50**
~~**J**~~	~~**60**~~



What is the right algorithm?

- on each machine SPT order

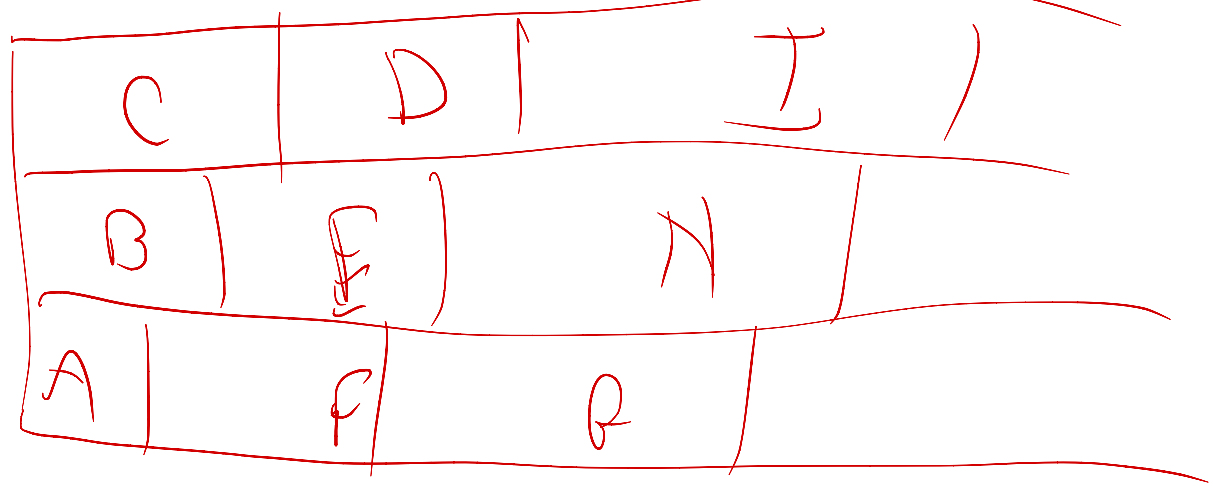
$$\sum C_j = 3p_A + 2p_D + p_G + 3p_B + 2p_E + p_H + 3p_C + 2p_F + p_I$$

$$= 3(p_A + p_B + p_C) + 2(p_D + p_E + p_F) + (p_G + p_H + p_I)$$

Average Completion Time on Multiple Machines

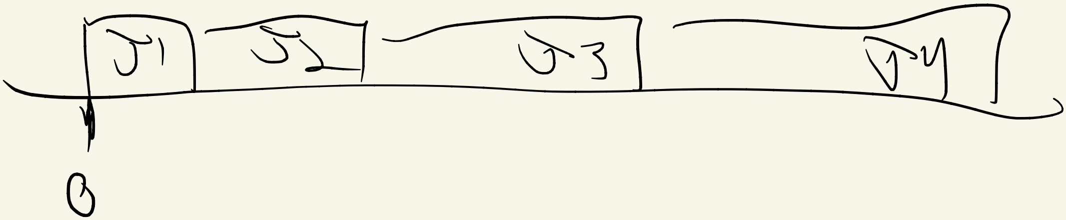
$P||\sum C_j$

j	p_j
A	**1**
B	**3**
C	**4**
D	**5**
E	**6**
F	**9**
G	**12**
H	**20**
I	**50**
~~**J**~~	~~**60**~~



What is the right algorithm?

$1 || \sum C_j$ SPT



$$\sum C_j = C_1 + C_2 + C_3 + C_4$$

$$= p_1 + (p_1 + p_2) + (p_1 + p_2 + p_3) + (p_1 + p_2 + p_3 + p_4)$$

$$= 4p_1 + 3p_2 + 2p_3 + 1p_4$$

$1 $5 $10 $100

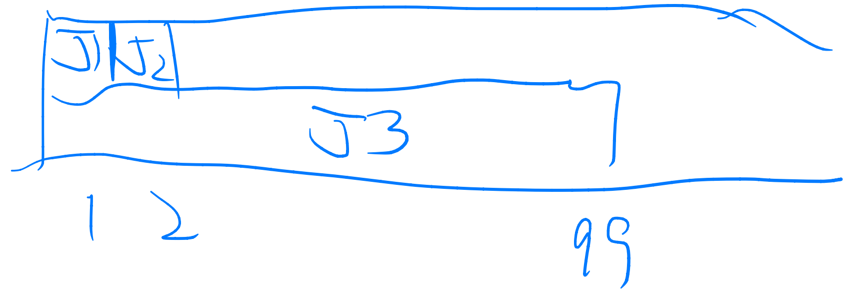
Average Completion Time on Multiple Machines

- $P||\sum C_J$ – **SPT is optimal.**
- $P||\sum w_j C_j$ – **Is WSPT optimal?**

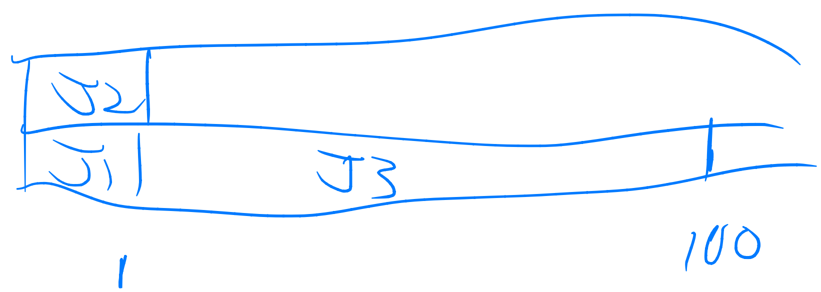
2 machines   Smith's rule

Example

j	w_j	p_j	w_j/p_j
1	**1**	**1**	1
2	**1**	**1**	"
3	**100**	**99**	100/99



$\sum w_j C_j = 1(1)\ 1(2) + 100(99) = 9903$



$1(1) + 1(1) + 100(100) = 10002$

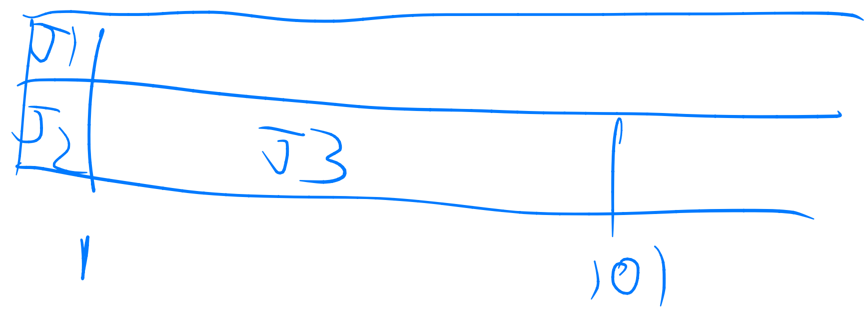
Average Completion Time on Multiple Machines

- $P||\sum C_J$ – **SPT is optimal.**
- $P||\sum w_j C_j$ – **Is WSPT optimal?**

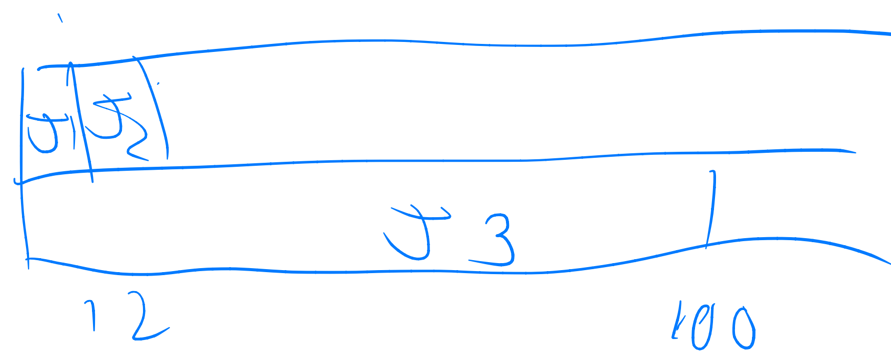
Smiths rule

Example

j	w_j	p_j	w_j/p_j
1	**1**	**1**	1
2	**1**	**1**	1
3	~~**100**~~ 99	~~**99**~~ 100	.99



$1(1) + 1(1) + 99(101) = 9999 + 2$
$= 10001$



$1(1) + 2(1) + 99(100) = 9903$

Average Completion Time on Multiple Machines

- $P||\sum C_J$ – **SPT is optimal.**
- $P||\sum w_j C_j$ – **Is WSPT optimal?**

Example

j	w_j	p_j
1	**1**	**1**
2	**1**	**1**
3	**100**	**99**

- $P||\sum w_j C_j$ **is NP-complete.**
- **WSPT is a** $(1+\sqrt{2})/2$ **-approximation for** $P||\sum w_j C_j$

≈ 1.22

$R||\Sigma C_j$

- Can be solved as a matching problem.
- Left side node for each job j
- Right hand side node for the k th from last job on machine i

suppose you knew

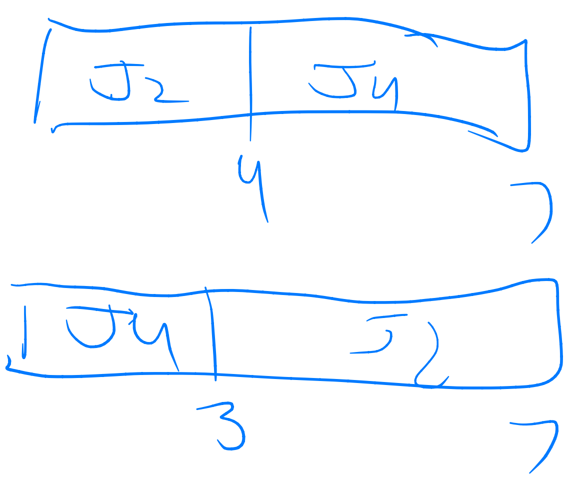
M_1 J_2 J_4

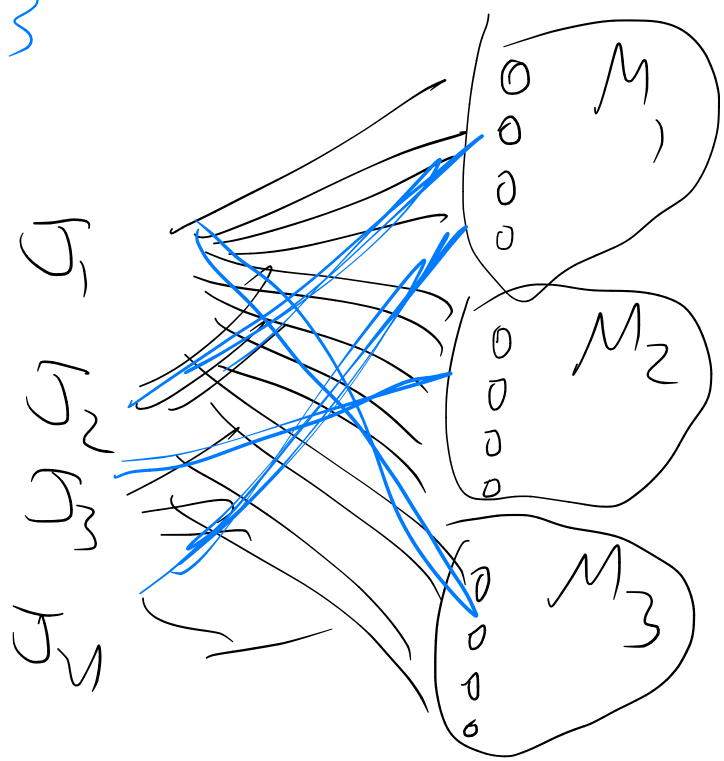
M_2 J_3

M_3 J_1

Example

	J_1	J_2	J_3	J_4
M_1	**6**	**4**	∞	**3**
M_2	**7**	**5**	**2**	**3**
M_3	**3**	**8**	**5**	**3**





$R || \Sigma C_j$

- Can be solved as a matching problem.
- Left side node for each job j
- Right hand side node for the k th from last job on machine i

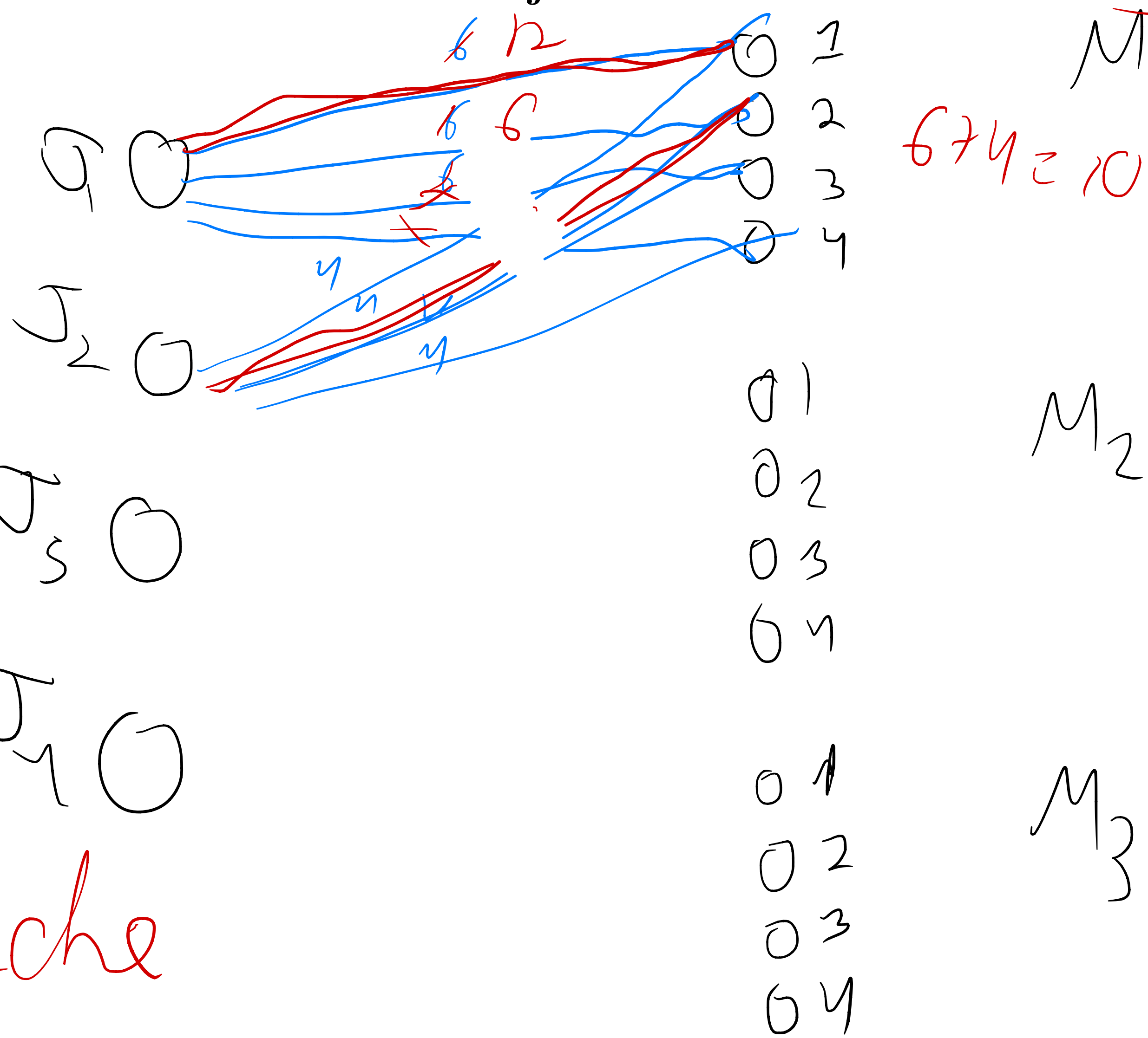
J_1	J_2

6 10

6+10=16

Example

	J_1	J_2	J_3	J_4
M_1	**6**	**4**	∞	**3**
M_2	**7**	**5**	**2**	**3**
M_3	**3**	**8**	**5**	**3**



problem don't know how many jobs on each meche

$R||\Sigma C_j$

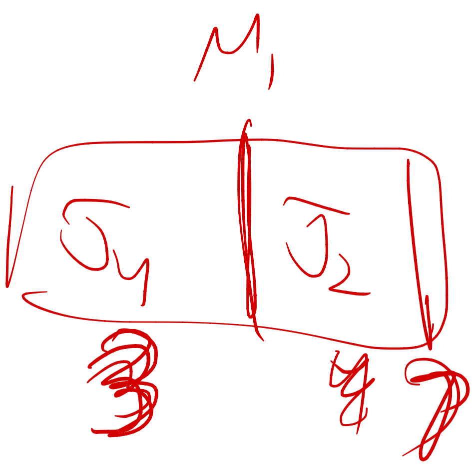
$cost(j, M_i, k)$

$k \cdot p_{ij}$

- **Can be solved as a matching problem.**
- **Left side node for each job** j
- **Right hand side node for the** k **th from last job on machine** i

Example

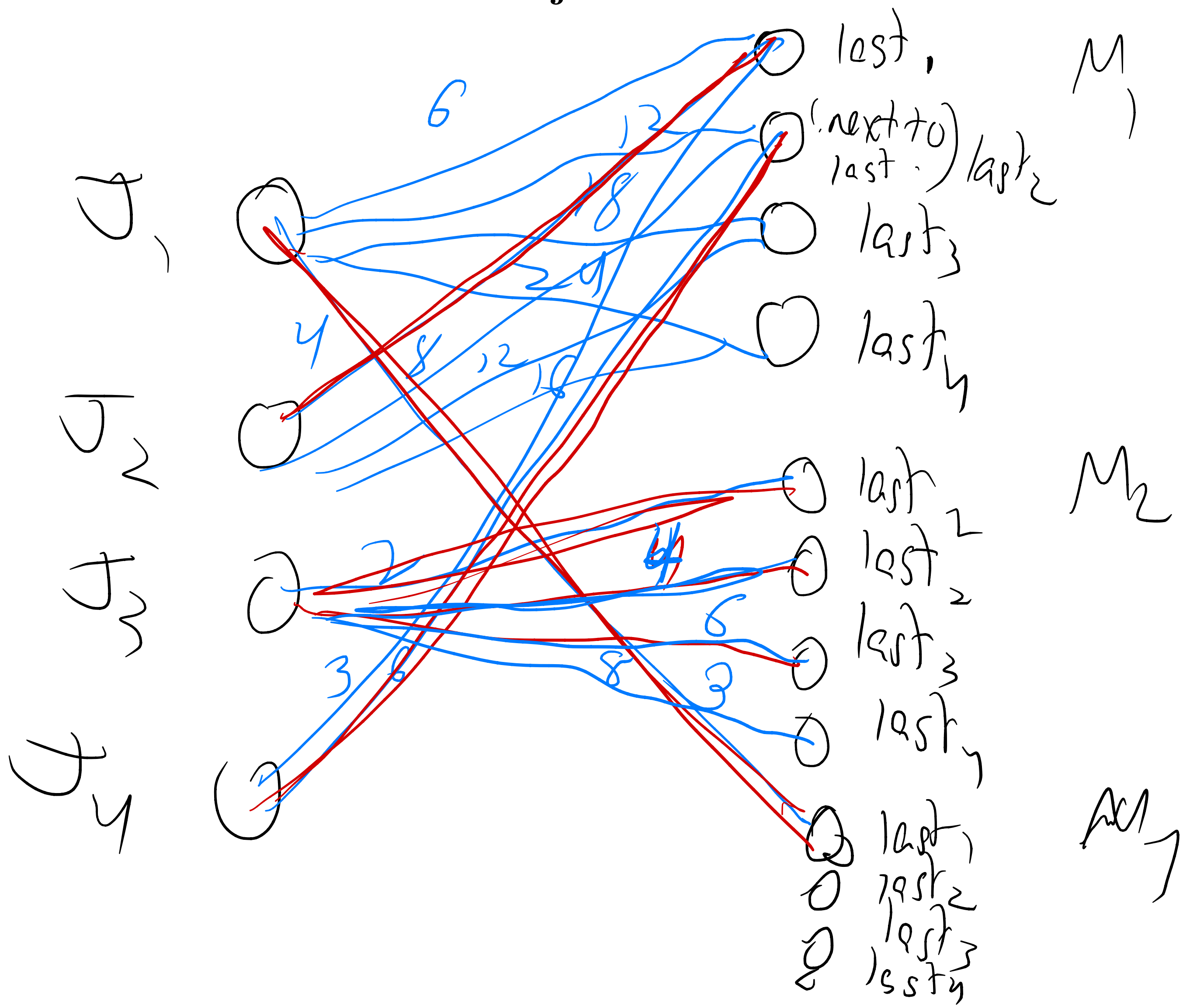
	J_1	J_2	J_3	J_4
M_1	**6**	**4**	∞	**3**
M_2	**7**	**5**	**2**	**3**
M_3	**3**	**8**	**5**	**3**

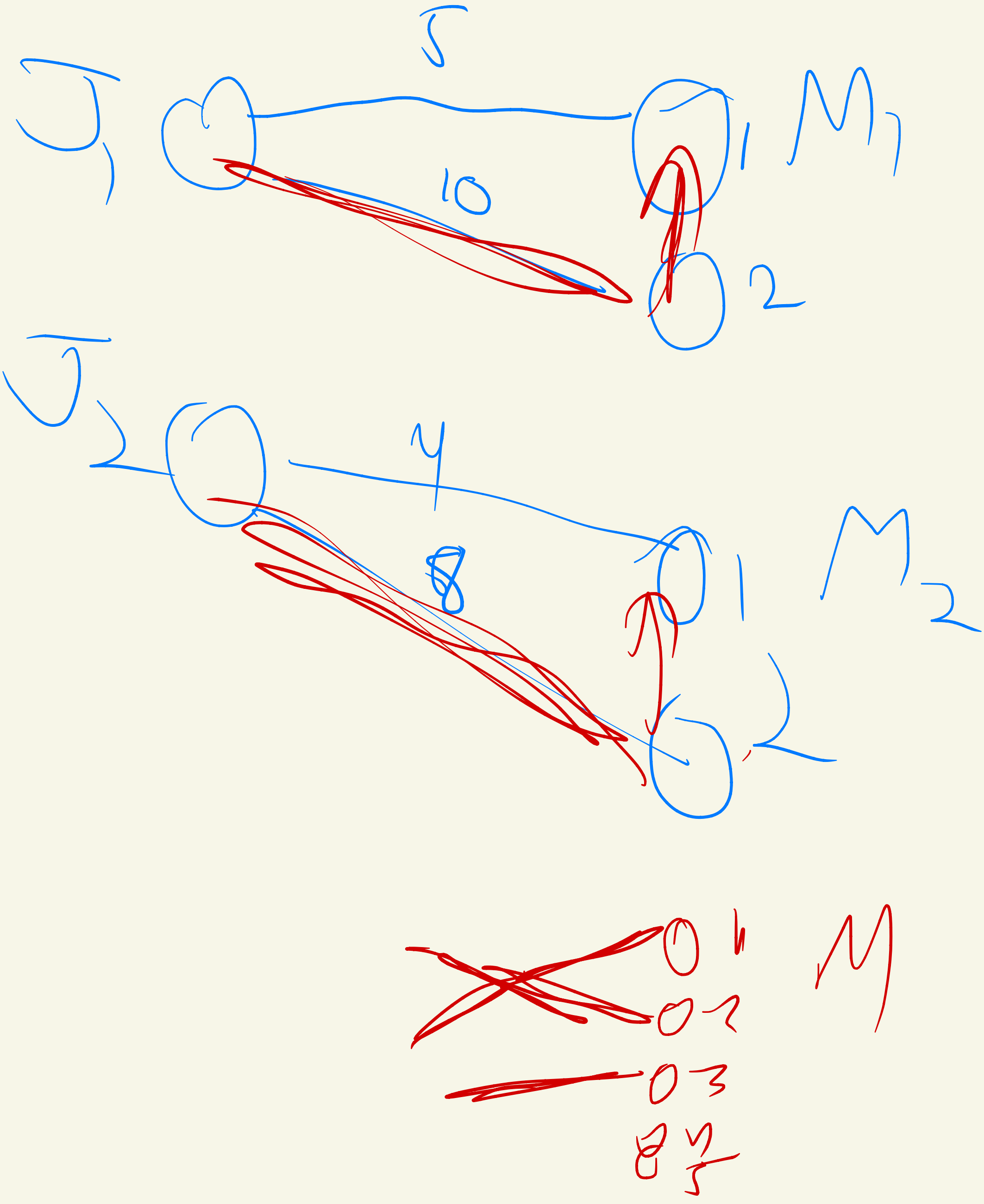


$3+7=10$

$3+(3+4)$

$= 6+4$





J1
5
10
M1
J2
4
8
M2
M

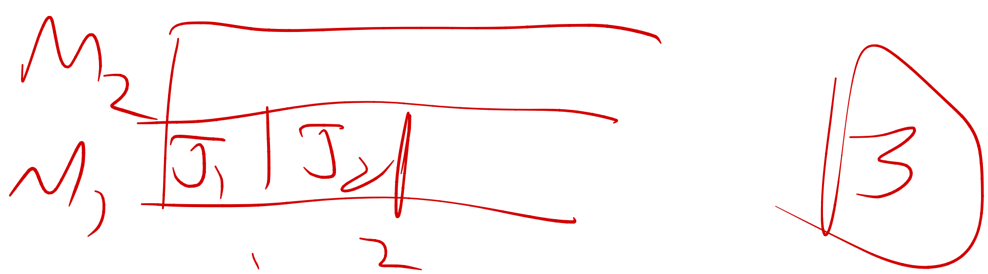
LP for the matching problem

Variable $x_{ijk} = 1$ **if** j **is the** k **th from last job on** M_i

$$
\begin{aligned}
\min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} k p_{ij} x_{ijk} & \\
\textbf{s.t.} & \\
\textbf{Each job runs} & \\
\sum_{i=1}^{m} \sum_{k=1}^{n} x_{ijk} = 1 & \qquad j = 1 \ldots n \\
\textbf{Each machine/slot has at most 1 job} & \\
\sum_{j=1}^{n} x_{ijk} \leq 1 & \qquad i = 1 \ldots m; k = 1 \ldots n \\
x_{ijk} \in \{0, 1\} & \quad i = 1 \ldots m; j = 1 \ldots n; k = 1 \ldots n
\end{aligned}
$$

- **Note that the may be unforced idleness e.g.**

	J_1	J_2
M_1	**1**	**1**
M_2	**10**	**10**



$Q|\mathrm{pmtn}|\Sigma C_j$

- **Algorithm is SRPT-FM. Shortest Remaining Processing Time on the Fastest Machines.**

j	p_j	rem		
A	~~32~~		~~$28\frac{1}{2}$~~	$22\frac{1}{2}$
B	20	~~19~~	~~12~~	0
C	16	~~14~~	0	
D	4	0		

$\frac{45}{2} \cdot \frac{1}{4} = \frac{45}{8}$

- **What about preemption in other models?**
- **P – doesn't help**
- **R – NP-complete**