10/9/13

## Spanning Tree Algorithm

After adding edges $e_1, e_2, \ldots, e_k$ we have components $C_1, C_2, \ldots, C_l$

Choose $C_i$ and add shortest edge with one end in $C_i$.

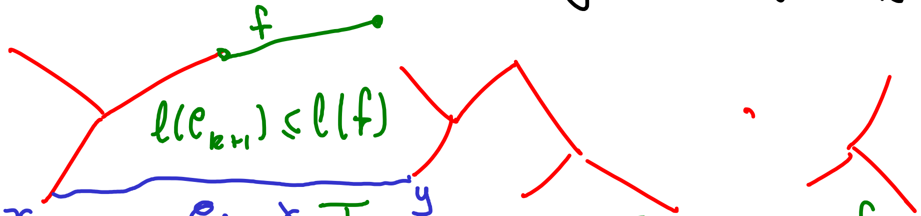
**Claim** at all times $\exists$ a minimum length tree that all the edges chosen so far.

By induction: true when 0 edges chosen

Suppose we have $k$ chosen edges $e_1, e_2, \dots, e_k \subseteq T \leftarrow$ min. length.

(i) $e_{k+1} \in T$ ✓

(ii) $e_{k+1} \notin T$




Replace $T$ by $T + e_{k+1} - f$

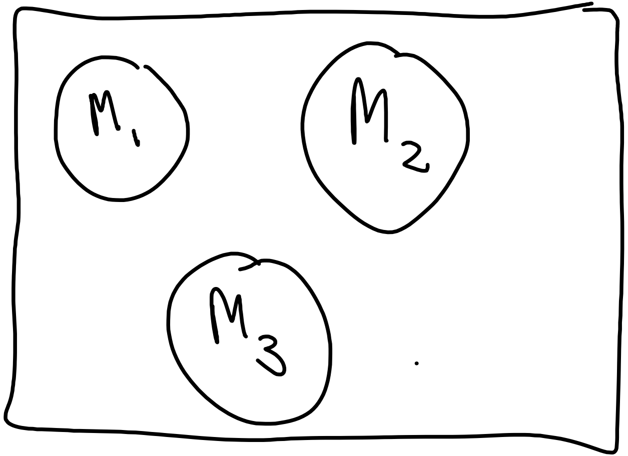
$T + e_{k+1}$ contains a cycle

# Job Shop Scheduling

Jobs

$J_1, J_2, \ldots, J_n$

Each job has to go through a sequence of machines to be finished.




Must work out the order in which jobs go through to minimise some objective. —

Can be NP-hard

## Example 1

$$1 \mid \quad \mid \sum_j w_j C_j$$

#machines

restrictions

objective

All jobs arrive at time 0.

$C_j$ = completion time of job $j$

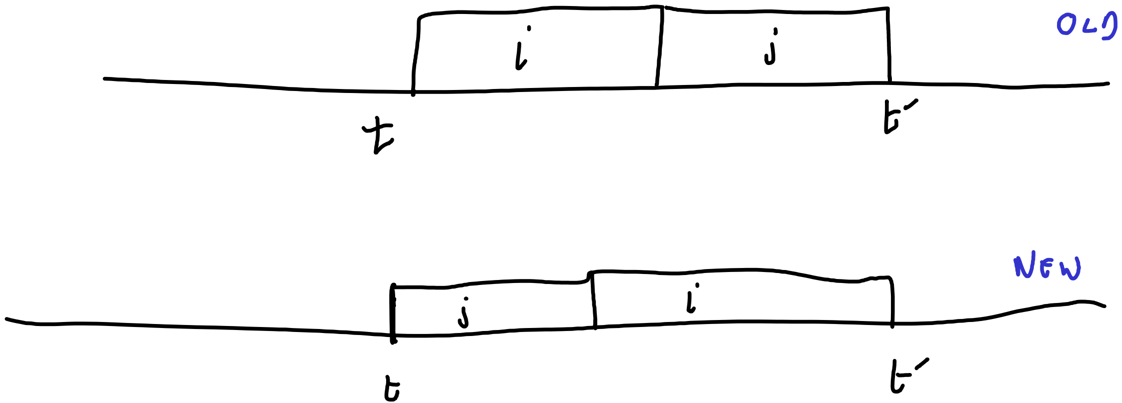
$p_j$ = processing time of job $j$

$w_j$ = a weight

optimal order : $\frac{p_1}{w_1} \le \frac{p_2}{w_2} \le \cdots$

what is the best order

$$\frac{p_i}{w_i} > \frac{p_j}{w_j}$$




$$w_i p_j - w_j p_i < 0.$$

NEW COST - OLD COST $= w_i(t+p_j+p_i) + w_j(t+p_j) - w_i(t+p_i) - w_j(t+p_i+p_j)$