§13 Performance Guarantees of simple policies for the expected makespan

Determining an optimal policy is NP-hard in general

Therefore: Determine "good" policies fast

Question: How good are they?

Def <u>ϱ-approximation algorithm A</u> for a minimization problem

- runs in polynomial time
- finds for every instance I a feasible solution $A(I)$
- fulfills $A(I) \leq \varrho \cdot OPT$

$A(I)$: value of $A(I)$

ϱ: "performance guarantee"

OPT: optimal value

Example: TSP

Instance: I : complete graph G with edge weights w_{ij} fulfilling triangle inequality

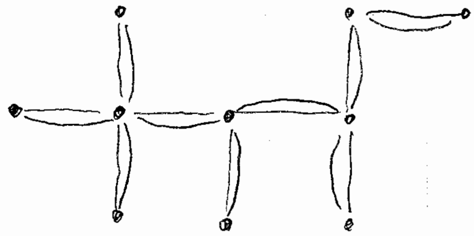
feasible solution: tour through all vertices

Problem is NP-hard

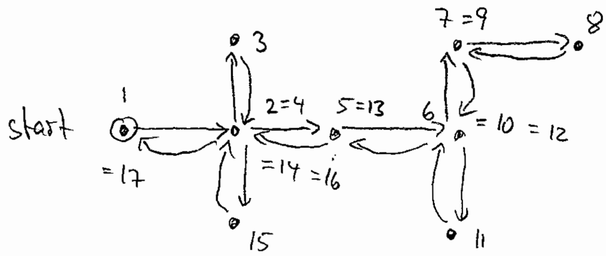
An approximation algorithm MST (min. spanning tree)

- Find a minimum spanning tree T of G
- double all edges of T $\longrightarrow T^*$
- find a Eulerian tour in T^*
- replace subtours of already visited vertices by the direct edge between the endpoints

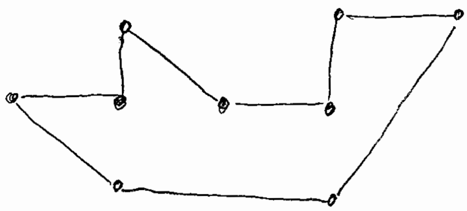
Expl: (distance = Euclidean distance)



T^*



Eulerian tour T_E in T^*



replacing visited subtours by edges → tour MST(I)

- algorithm is polynomial
- constructs a tour
- $MST(I) \le 2 \cdot OPT$

since
$$\left.\begin{array}{l} \text{length}(T) \le OPT \\ MST(I) \underset{\substack{\uparrow \\ \text{triangle ineq.}}}{\le} 2 \cdot \text{length}(T) \end{array}\right\} \Rightarrow MST(I) \le 2 \cdot OPT$$

<u>In our case:</u>

$I \mathrel{\hat{=}}$ scheduling problem

feasible solution $A(I) \mathrel{\hat{=}}$ policy Π

value of $A(I) \mathrel{\hat{=}}$ expected cost $E(\kappa^{\Pi})$ of Π

Can give performance guarantees only for machine scheduling models

13.1 THEOREM: Let G be arbitrary, $\mathcal{F} \hat{=}$ m machines and $\kappa = C_{max}$. Then, for every static priority rule Π and every vector x of processing times

$$C_{max}^{\Pi}(x) \leq \left(1 + \frac{m-1}{m}\right) \cdot OPT(x)$$

The bound is asymptotically tight

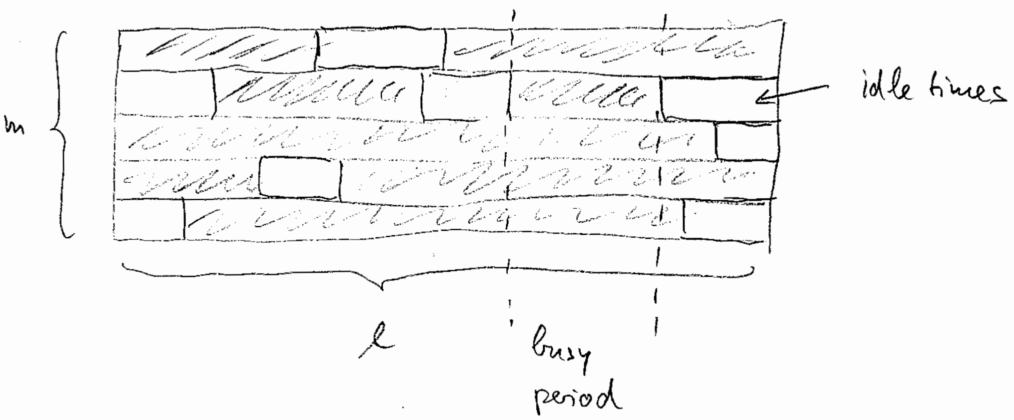
13.2 REMARK: Bound $1 + \frac{m-1}{m}$ is achieved "pointwise", i.e. for every x. Thus also

$$\mathbb{E}(C_{max}^{\Pi}) \leq OPT \quad (= \text{optimal expected makespan})$$

for every joint distribution of processing times

<u>Proof of Thm 13.1:</u>

Consider $\Pi[x]$ as an $m \times \ell$ rectangle with ℓ = makespan



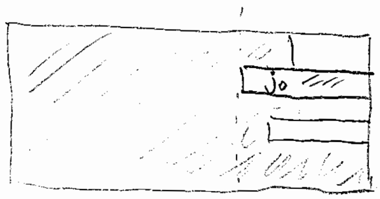
$\left.\begin{array}{l} I_k \text{ be the idle time on machine } k \\ B_k \text{ be .. busy ... } k \end{array}\right\} \Rightarrow \sum_{k=1}^{m} (|I_k| + |B_k|) = m \cdot \ell$

no idle time $\Rightarrow$ $\Pi[x]$ is optimal $\Rightarrow$ statement.

So suppose there is idle time.

Consider job j_0 that ends last

<u>CASE 1</u>: all idle times are parallel to j_0



all machines busy

$$\Rightarrow \sum_{k=1}^{m} |I_k| \leq (m-1)\, x_{j_0} \leq (m-1) \cdot OPT$$

$$\Rightarrow \ell \cdot m = \sum_{k=1}^{m} |I_k| + \sum_{k=1}^{m} |B_k| \leq (m-1) \cdot OPT + m \cdot OPT$$

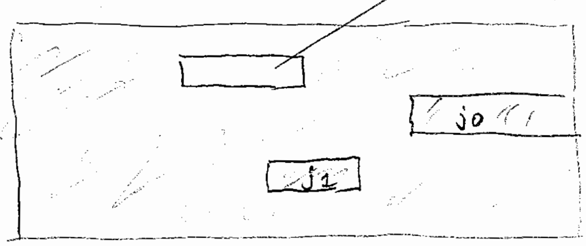
$$\Rightarrow \ell \leq \left(\frac{m-1}{m} + 1 \right) OPT$$

CASE 2: there are idle times before the start of j_0

Consider last such idle time.

Why was j_0 not started in that idle time?

Because a predecessor j_1 of j_0 was busy at the end of that idle time



$\Rightarrow$ (inductively) there is a chain

$$j_t <_G j_{t-1} <_G \dots <_G j_1 <_G j_0$$

such that every idle time is covered by the processing of that chain

(i.e. for every $t \in \bigcup I_k$, there is a job from the chain that is busy at t)

$$\Rightarrow \sum_{k=1}^{m} |I_k| \leq (m-1)(\text{length of the chain}) \leq (m-1) \cdot OPT$$

$$\Rightarrow (\text{as in CASE 1}) \quad \ell \leq \left(\frac{m-1}{m} + 1\right) OPT \quad \square$$

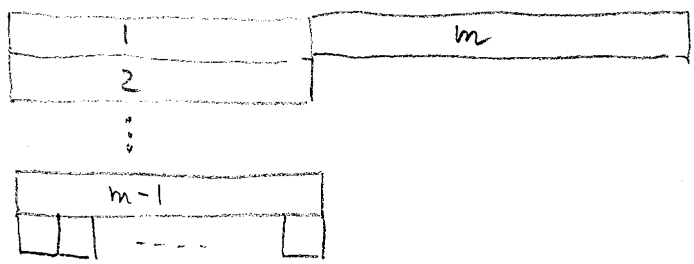
13.2 EXAMPLE (Tightness of the bound)

$V = \{1, \ldots, m, m+1, \ldots, 2m-1\}$, no precedence constraints

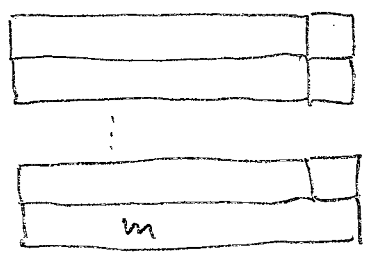
$$x = (\underset{1}{m-1}, \underset{2}{m-1}, \ldots, \underset{m-1}{m-1}, \underset{m}{m}, \underset{m+1}{1}, 1, \ldots, \underset{2m-1}{1})$$

priority list: $1 < 2 < \ldots < m-1 < m+1 < \ldots < 2m-1 < m$

$\Pi[x]$:



OPT(x)



$$\Rightarrow \frac{\Pi[x]}{OPT} = \frac{m-1+m}{m} = 1 + \frac{m-1}{m} \quad \square$$