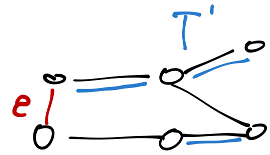
Analysis of Kruskal's algorithm

Proof of lemma 3.1: Let $e \in E$ and $f \in C_T(e)$. Assume by contradiction that $d(e) < d(f)$. Let $T' \subseteq T$ be the set of edges that were added before f. Because $C_T(e)$ is unique cycle in T, $T' \cup \{e\} \subseteq T \cup \{e\} \setminus \{f\}$ contains no cycle. So greedy algorithm would have chosen e instead of f. $\lightning$



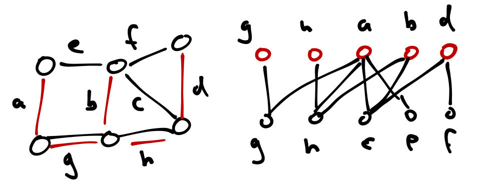
Proof of Theorem 3.3:

plan: $\sum_{f \in T} d(f) \leq \sum_{f \in T} d(e_f)$ ← $e_f \in T^*$ with $f \in C_T(e_f)$

Let T^* be an MST. Construct bipartite graph

$H = (T \cup T^*, \tilde{E})$

with $\tilde{E} := \{\{e,f\} : e \in T^*, f \in C_T(e)\}$



Claim: H contains a matching M covering T, $M = \{\{f, e_f\} : f \in T, e_f \in T^*\}$ with $f \in C_T(e_f)$

By swap optimality: $d(e_f) \geq d(f) \ \forall f \in T$

Thus $\sum_{f \in T} d(f) \leq \sum_{f \in T} d(e_f) \leq \sum_{e \in T^*} d(e) = OPT$. $\square$

M covers T

$\boxed{f \neq f' \Rightarrow e_f \neq e_{f'}}$

Proof of Claim We use Hall's Theorem:

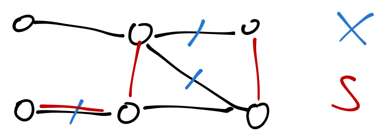
$H = (A \cup B, \tilde{E})$ contains a matching covering A iff $|\Gamma(X)| \geq |X|$ for all $X \subseteq A$. $\left(\Gamma(X) := \{b \in B : \{a,b\} \in \tilde{E} \text{ for some } a \in X\}\right)$

Let $X \subseteq T$. $T \setminus X$ contains no cycles and T^* connects V.

$\Rightarrow \exists\, S \subseteq T^* : T \setminus X \cup S$ is a spanning tree

$|T\setminus X| + |X| = |T| = |V| - 1 = |T \setminus X \cup S| \leq |T\setminus X| + |S| \Rightarrow |X| \leq |S|$

We show $S \subseteq \Gamma(X)$. Let $e \in S$. The tree $T \setminus X \cup S$ does not contain a cycle. In particular, it does not contain $C_T(e)$, i.e., there is $f \in C_T(e) \setminus (T \setminus X \cup S) \subseteq X$. $\Rightarrow e \in \Gamma(X)$ □



Scheduling on Parallel Identical Machines ($P||C_{max}$)

$$load_\sigma(i) := \sum_{j : \sigma(j) = i} p_j$$

Two lower bounds

Lemma $OPT \geq \max_{j \in [n]} p_j$ **Lemma** $OPT \geq \frac{1}{m} \sum_{j=1}^{n} p_j$

Analysis of Local Search for $P||C_{max}$ (Thm. 3.4)

Let $i^* \in \operatorname{argmax}_{i \in [m]} load_\sigma(i)$ and let j^* with $\sigma(j^*) = i^*$.

By local optimality: $load_\sigma(i) + p_{j^*} \geq load_\sigma(i^*) \quad \forall\, i \in [m]$

$$\Rightarrow \sum_{j \in [n]} p_j = \sum_{i \in [m]} load_\sigma(i) \geq m \cdot (load_\sigma(i^*) - p_{j^*})$$

Hence: $ALG = \underbrace{load_\sigma(i^*) - p_{j^*}}_{\leq \frac{1}{m} \sum_{j \in [n]} p_j} + \underbrace{p_{j^*}}_{\leq \max_{j \in [n]} p_j} \leq 2\, OPT$

*Running time: Needs a slightly different variant, see book. □

Analysis of List Scheduling (Thm. 3.5)

Let $i^* \in \operatorname{argmax} \mathrm{load}_\sigma(i)$ and let $j^* \in [n]$ be the last job that was assigned to i^*.

When j^* was assigned to i^*, the load of i^* was minimal.

Thus $\mathrm{load}_\sigma(i^*) - p_{j^*} \leq \mathrm{load}_\sigma(i)$ for all $i \in [m]$.

Rest as before. $\square$

Analysis of LPT List Scheduling (Thm. 3.6)

Let $i^* \in \operatorname{argmax} \mathrm{load}_\sigma(i)$ and let j^* be the last job assigned to i^*. W.l.o.g.: $j^* = n$ (removing jobs with $p_j \leq p_{j^*}$ does not change ALG and does not increase OPT).

If $p_n = \min_{j \in [n]} p_j > \frac{1}{3}\mathrm{OPT}$ then $\mathrm{ALG} = \mathrm{OPT}$ (Exercise 2.2).

If $p_n \leq \frac{1}{3}\mathrm{OPT}$, then

$$\mathrm{ALG} = \mathrm{load}_\sigma(i^*) \underset{\text{as before}}{\leq} \frac{1}{m}\sum_{j \in [n]} p_j + p_n \leq \mathrm{OPT} + \frac{1}{3}\mathrm{OPT}. \quad \square$$

Greedy Algorithm for k-CENTER

Proof of Thm. 3.7: For $v \in V$, let $c^*(v) \in \operatorname{argmin} d(v, S^*)$.

Case 1: $\forall w \in S^* \ \exists v \in S : c^*(v) = w$

Let $u \in V$ and $w = c^*(u)$. Let $v \in S$ with $c^*(v) = w$.

Then $d(u, S) \leq d(u, v) \leq d(u, w) + d(w, v) \leq 2\,\mathrm{OPT}$.

Case 2: $\exists v, v' \in S : c^*(v) = c^*(v') =: w$

Then $\mathrm{ALG} \underset{\text{by greedy choice}}{\leq} d(v, v') \leq d(v, w) + d(v', w) \leq 2\,\mathrm{OPT}$. $\square$