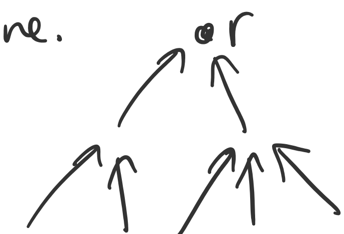
Minimum-Weight Arborescence

arcs

Def: Let $r \in V$ be a root vertex in a directed graph $G=(V,A)$.
An r-arborescence is a subgraph $T \subseteq A$ s.t.

① Every vertex has a directed path in T to the root r,

② Each vertex except r has exactly one outgoing edge, r has none.



Remark: T forms a spanning tree in the undirected sense.

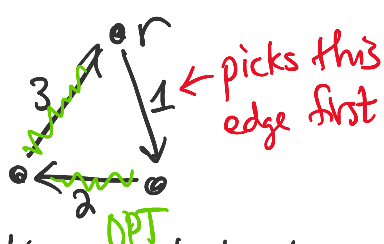
Remark: Can check if r-arborescence exists: run "reverse DFS/BFS" starting from root.

Minimum Weight Arborescence

Input: weighted directed graph $G=(V,A)$ and root $r \in V$

Output: r-arborescence of minimum total weight.

Attempt: Greedy algorithm (Kruskal)



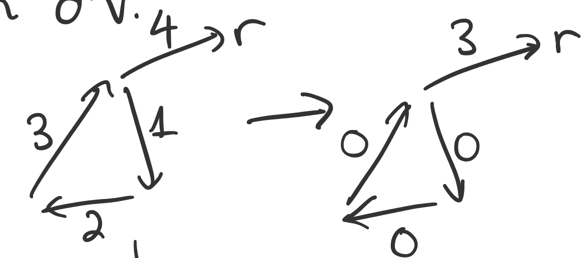
Edmonds' Algorithm: "Boruvka-like contraction with backtracking"

Def: For each $v \in V$, let $\partial^+ v \subseteq A$ be all arcs leaving vertex a.

Let $M_G(v) = \min_{a \in \partial^+ v} w(a)$ be the minimum weight arc leaving a.

① Pre-processing step: for each $v \in V$, subtract $M_G(v)$ weight from

① Preprocessing step: for each $v \in V$, subtract $M_G(v)$ weight from all arcs in $\partial^+ v$.



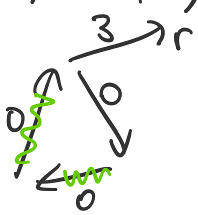
Let G' be the new graph.

Claim: T is min-weight arborescence in $G \iff$

T " " " " G'.

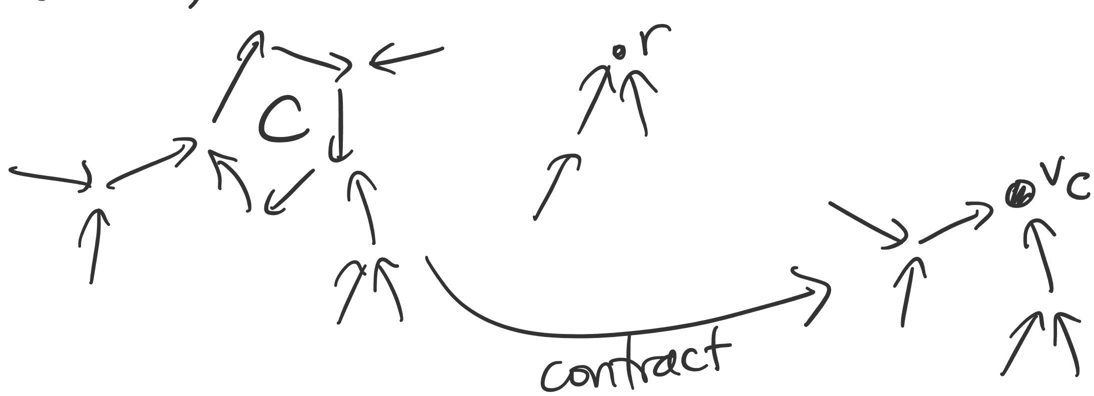
Proof: Any arboresence T contains exactly one arc in each $\partial^+ v$ $(v \neq r)$, so $w_{G'}(T) = w_G(T) - \underbrace{\sum_{v \neq r} M_G(v)}_{\text{static}}$.

② For each vertex $v \neq r$, pick an arbitrary 0-weight edge leaving v.



If the edges form an arborescence, then done.

Otherwise, there are cycles.



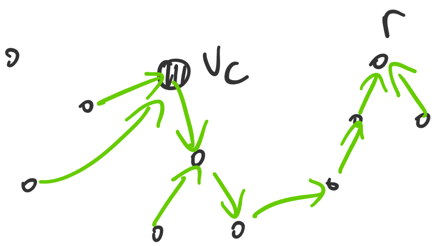
Contract one cycle, remove parallel edges (keep only cheapest)

Contract <u>one</u> cycle, remove parallel edges (keep only cheapest)

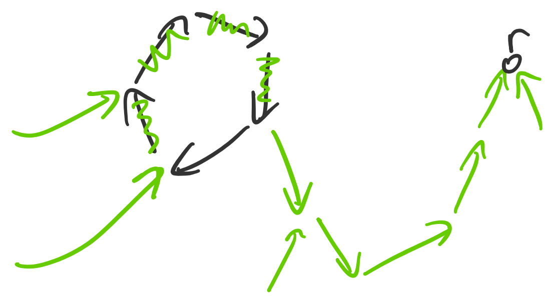
Let G'' be new graph.

Claim: $OPT(G') = OPT(G'')$

Proof: $OPT(G') \leq OPT(G'')$: Let T'' be min-weight arborescence in G''.



Expand $v_c \to C$:



Add all but one arc of C into $T'' \to T'$

C has 0-weight edges $\Rightarrow w_{G''}(T'') = w_{G'}(T')$.

$OPT(G'') \leq OPT(G')$: Let T' be a min-weight arborescence in G'. Contract $C \to v_c$ in T'.

Contraction $\Rightarrow$ each vertex can still reach r.

So exists subgraph T'' that is arborescence of G'', and $w_{G'}(T') \leq w_{G''}(T'')$.

Run time: ① + ② in $O(m)$ time, # vertices decreases

$\Rightarrow O(mn)$ overall.

Remark: current best $O(m \log\log n)$.

Alternate Proof of Correctness using LP

Linear Program

Input: n = # variables = dimension

m = # constraints

A = constraint matrix

b = constraint vector

c = objective vector

Objective: minimize $\underbrace{c^T x}_{= c \cdot x = \sum_{i=1}^{n} c_i x_i}$ s.t. $\underbrace{Ax \geq b \text{ and } x \geq \vec{0}}$.

if it satisfies the constraints.

Def: x is a feasible solution

Def: The dual LP is

$$\text{maximize } b^T y \text{ s.t. } A^T y \leq c \text{ and } y \geq 0.$$

Theorem (Weak Duality): If x is feasible solution of primal and y " " " dual,

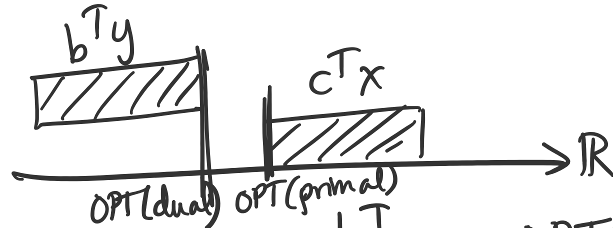
then $c^T x \geq b^T y$.

Proof: $c^T x \geq (A^T y)^T x \quad [A^T y \leq c,\ x \geq 0]$

$= y^T A x$

$\geq y^T b \qquad [Ax \leq b,\ y \geq 0]$

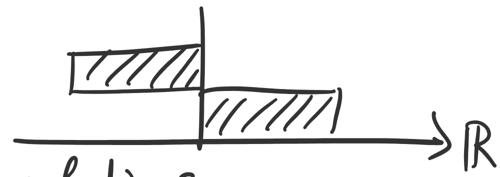
$= b^T y.$



In particular,

$$\mathrm{OPT(primal)} = \min_{x \text{ feasible}} c^T x \geq \max_{y \text{ feasible}} b^T y = \mathrm{OPT(dual)}.$$

If x, y feasible s.t. $c^T x = b^T y$



then x, y are optimal primal, dual solutions.

Strategy: ① encode algorithm output as x in min-weight arborescence LP.

② Construct y in dual LP s.t. $c^T x = b^T y$

$\Rightarrow x$ is optimal solution.

Min-weight arborescence LP

Encode each arborescence T as $x_T \in \{0,1\}^A$:

for each $a \in A$, $x_a = 1$ if $a \in T$

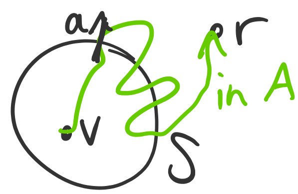
$x_a = 0$ if $a \notin T$.

$$\mathrm{OPT} = \min_{A \text{ arbor.}} w^T x_T.$$

Encode constraint "A is an arborence":

Encode constraint "A is an arborence":

Let $S \subseteq V \setminus \{r\}$, $v \in S$.



A is an arborescence

$\Rightarrow$ v can reach r in A

$\Rightarrow$ $\exists$ arc $a \in A$ out of S

Let $\partial^+ S \subseteq A$ be arcs out of S in G, then

$$\partial^+ S \cap T \neq 0 \quad \forall S \subseteq V \setminus \{r\}.$$

$$\Updownarrow$$

$$\sum_{a \in \partial^+ S} x_a \geq 1$$

Integer Program: minimize $\displaystyle\sum_{a \in A} w(a) x_a$

$$\text{s.t. } \sum_{a \in \partial^+ S} x_a \geq 1 \quad \forall S \subseteq V \setminus \{r\}, |S| > 1$$

$$\sum_{a \in \partial^+ v} x_a = 1 \quad \forall v \in V \setminus \{r\}$$

$$x_a \in \{0,1\}^A$$

Claim: OPT(MWA) = OPT(IP).

LP Relaxation: use $x_a \in [0,1]^A$ instead.

Claim: $\text{OPT(LP)} \leq \text{OPT(IP)} = \text{OPT(MWA)}$.

LP Dual: maximize $\displaystyle\sum_{S \subseteq V \setminus \{r\}} y_S$

s.t. $\displaystyle\sum_{S:\, a \in \partial^+ S} y_S \leq w(a) \quad \forall a \in A$

$y_S \geq 0 \quad \forall S \subseteq V \setminus \{r\},\ |S| > 1$

($y_{\{r\}}$ unconstrained)