

Network Simplex

► Algorithm:

- ① compute the dual vector p
- ② compute the reduced costs \bar{c} ; STOP if $\bar{c} \geq 0$
- ③ $B :=$ set of backward edges of the cycle formed by an edge e with $\bar{c}_e < 0$;
STOP if $B = \emptyset$
- ④ push flow around this cycle and update basis and basic solution

► Degeneracy: if some tree edge has flow 0

- Θ^* could be 0
- tree changes, but not the flow
- cycling can happen!
- anticycling rules!

► Running time (of one iteration):

- compute p : $O(n)$
- compute \bar{c} : $O(m)$
- cycle: $O(n)$
- updating the flow: $O(n)$

$O(m)$

(since $m \geq n-1$)

Important: data structures that support access to edges & nodes of graph and trees in constant time

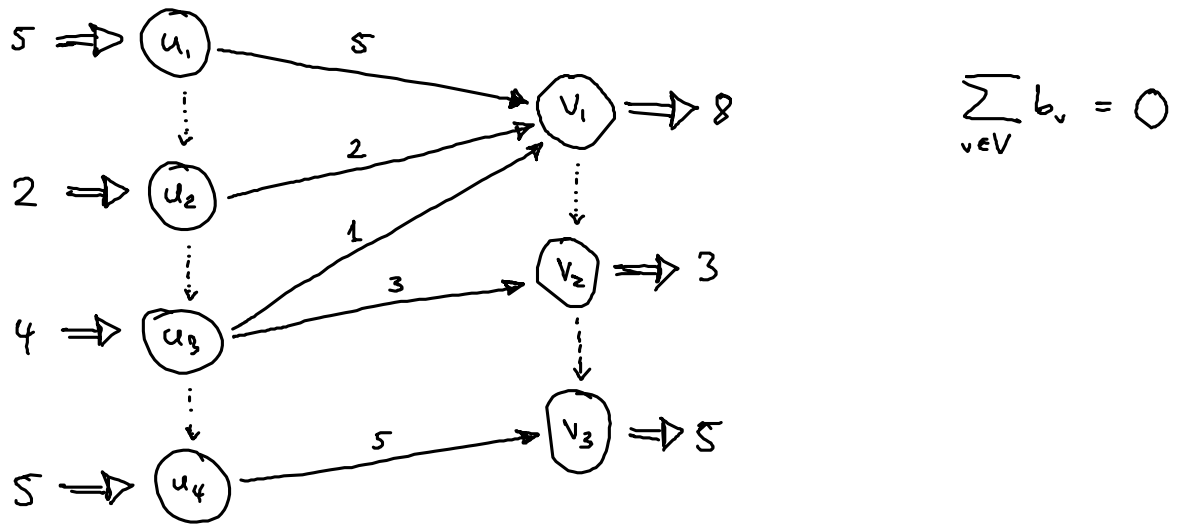
The number of iteration can be $\Omega(2^n)$, but is most often $O(n)$.

► Finding an initial basic feasible solution:

- Add auxiliary edges from each source to each sink
(if not already there) with cost $n \cdot \max\{c_e \mid e \in E\} + 1$

Let u_1, \dots, u_e be the source nodes and v_1, \dots, v_k the sink nodes.
 Put flow on all auxiliary edges "from top to bottom":

Ex.:



► Possible outcomes of the algorithm

- directed cycle with reduced cost < 0 (step ③)
 → unbounded problem
- optimal solution (step ②) with auxiliary edges
 → (original) problem is infeasible
- optimal solution (step ②) without auxiliary edges
 → problem is solved and output is an optimal primal solution together with an optimal dual solution!

► Capacitated problems:

Suppose we have bounds on the flow of an edge e :

$$0 \leq d_e \leq f_e \leq u_e \quad \forall e \in E$$

Partition the edges not in the tree into two sets D and U with

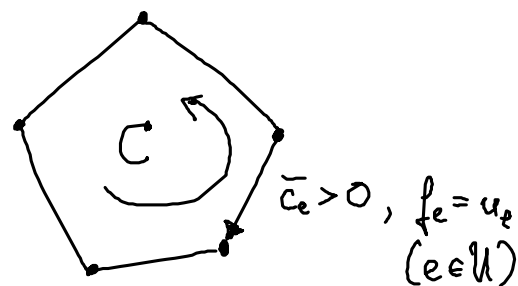
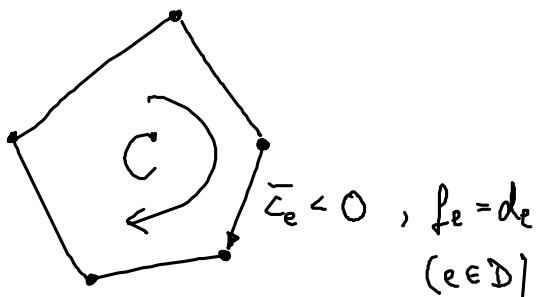
$$f_e = d_e \quad \forall e \in D, \quad f_e = u_e \quad \forall e \in U$$

Basis change: Compute \bar{c}_e as before for all $e \in D \cup U$. Then

Either } find an edge $e \begin{cases} \in D \\ \in U \end{cases}$ with $\begin{cases} \bar{c}_e < 0 \\ \bar{c}_e > 0 \end{cases}$.
Or }

This gives a cycle C .

Orient C such that e is a $\begin{cases} \text{forward} \\ \text{backward} \end{cases}$ arc



$F :=$ set of forward edges of C , $B :=$ set of backward edges of C

$$\Theta^* := \min \left\{ \min_{e' \in B} \{f_{e'} - d_{e'}\}, \min_{e' \in F} \{u_{e'} - f_{e'}\} \right\}$$

→ At least one edge e' will leave its flow value set either to $d_{e'}$ or $u_{e'}$.

If $e' \in T$ then e' leaves T and goes to either D or U and e replaces e' in T

If $e' = e$ then e changes from D to U or vice versa.

→ degeneracy: if $\Theta^* = 0$ then the basis changes, but not the flow

↑
T and either D or U