

## Recitation 9:

November 1, 2005

### Network Simplex Algorithm:

- Start with a basic feasible solution f (ass. to a tree)

- Compute dual vector  $p$  by solving

$$p_i - p_j = c_{ij} \quad \forall (i,j) \in T$$

$$p_n = 0$$

- Compute reduced costs for  $(i,j) \notin T$  (nonbasic edges) by

$$\bar{c}_{ij} = c_{ij} - (p_i - p_j)$$

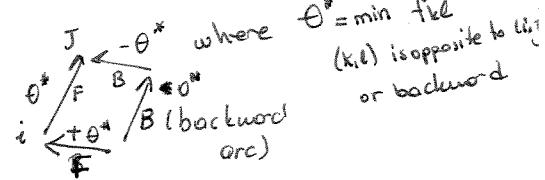
if all  $\bar{c}_{ij} \geq 0 \rightarrow$  optimal

else choose some  $(i,j)$  with  $\bar{c}_{ij} < 0 \rightarrow$  this edge  $(i,j)$  will enter the basis

- Find the cycle in  $T$  when  $(i,j)$  is added.

If  then unbounded

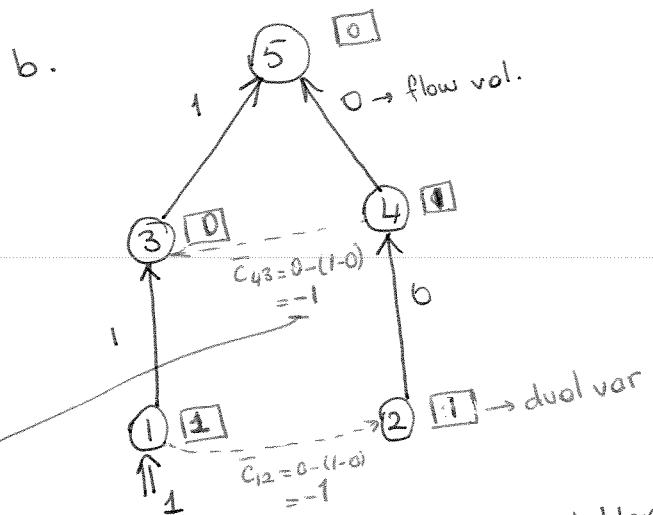
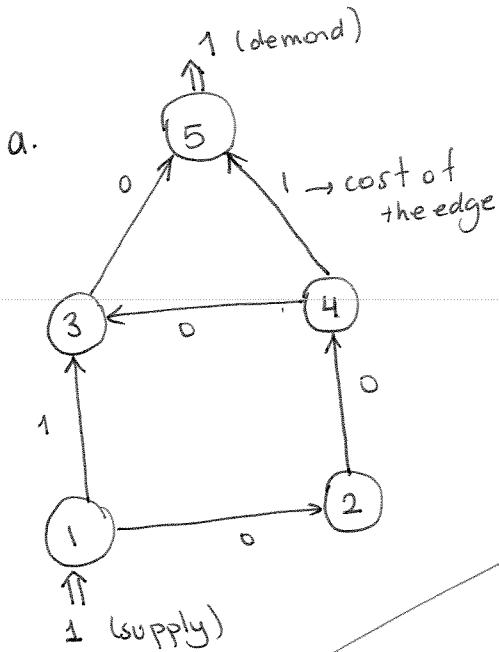
o.w.



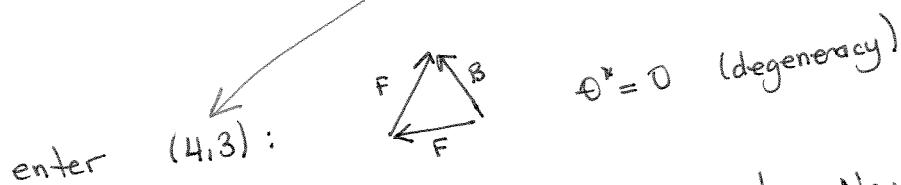
(forward arc)  
(backward arc)  
 $\theta^* = \min \text{fwd}$   
 $(k,i)$  is opposite to  $u_k$   
 $(i,k) \in \text{backward}$

- Remove the arc with 0 flow from the basis (arg min fwd  $(k,i) \in \text{backward}$ )

Ex:

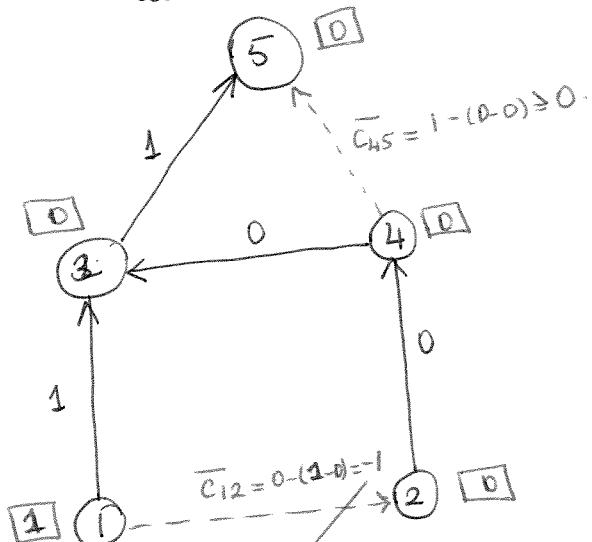


An initial feasible tree solution  
 $\text{cost} = 1$



c. New solution:

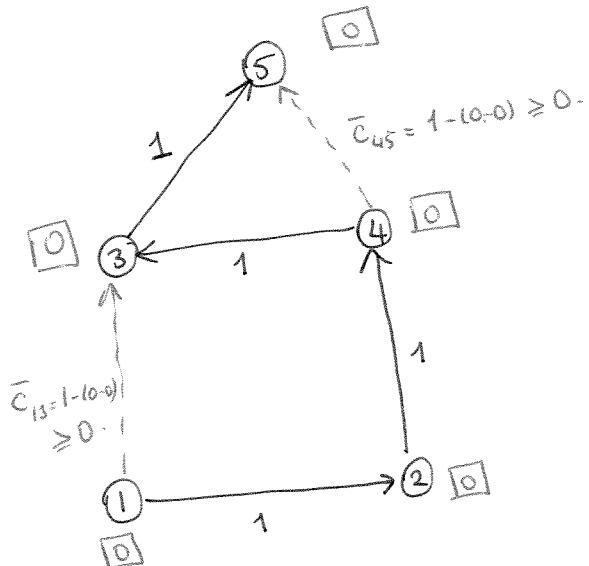
cost = 1



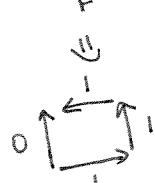
enter  $(1,2)$ :

d. Newer solution

cost = 0



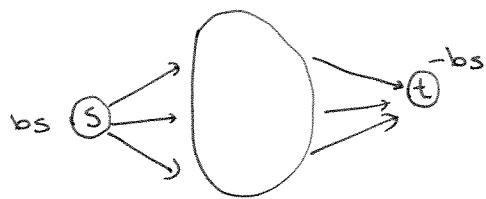
optimal solution!



## Ford-Fulkerson Algorithm:

Recall: Maximum Flow Problem:

$$\begin{array}{ll} \text{Max } & b_s \\ \text{s.t. } & Af = b \quad (\text{where } b_t = -b_s, b_i = 0 \text{ for } i \neq s, t) \\ & 0 \leq f_{ij} \leq u_{ij} \end{array}$$



Important Definition: Let  $f$  be a feasible flow vector.  
 An "augmenting path" is a path from  $s$  to  $t$  s.t.  
 $f_{ij} < u_{ij}$  for all forward arcs and  $f_{ij} > 0$  for all  
 backward arcs.  $(s \xrightarrow{f} \xleftarrow{B} \xrightarrow{f} \xleftarrow{B} \xrightarrow{f} t)$

Note that, if we have an augmenting path, we can increase  
 flow on forward arcs and decrease flow on backward arcs  
 without changing flow conservation constraints and obtain  
 a solution with better obj. funct. value.  
 The amount of flow that can be pushed is  
 $\delta(p) = \min \left\{ \min_{(i,j) \in F} (u_{ij} - f_{ij}), \min_{(i,j) \in B} f_{ij} \right\}$

So we have an algorithm for max. flow problem:

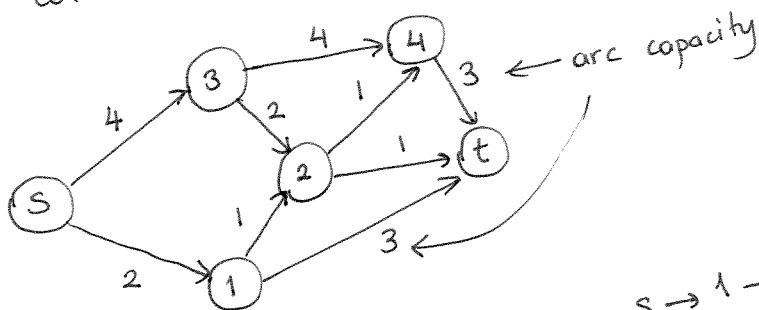
- Start with feasible flow  $f$  (usually  $f=0$ )
- Search for an augmenting path
- If no augmenting path can be found, terminate. (OPT)
- If an augmenting path is found, then
  - $\delta(p) < \infty$ , push  $\delta(p)$ , iterate
  - $\delta(p) = \infty$ , unbounded, terminate. (no path w/o cap. rest is found)

Note: If all  $w_{ij}$  are integer, arc flow values remain integer

and alg. terminates in finite # steps  
and alg. terminates in finite # steps  
and alg. terminates in finite # steps  
and alg. terminates in finite # steps

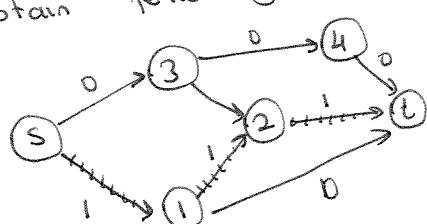
Note: Augmenting paths can be found or conclusion that none exists  
can be derived using a very simple labeling alg.

Ex:



$s \rightarrow 1 \rightarrow 2 \rightarrow t$ , all forward arcs with  $\delta=1$

a. Start  $f=0$ , we have augmenting path:  
Obtain following flow:

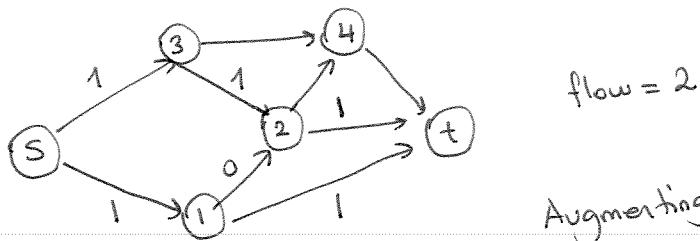


flow = 1

augmenting path:  $s \rightarrow 3 \rightarrow 2 \leftarrow 1 \rightarrow t$   
 $\delta = \min \{ \min(4, 2, 3), \min(1) \} = 1$

b. Augment on this path one unit flow to obtain:

c.



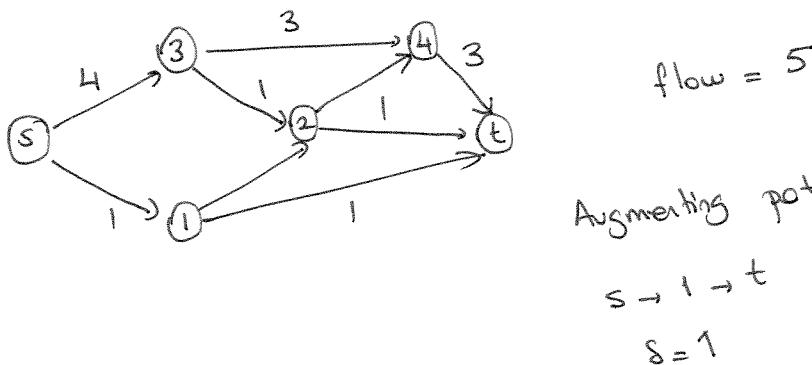
Augmenting path:

$$s \rightarrow 3 \rightarrow 4 \rightarrow t$$

$$s = 3$$

Augment:

d.



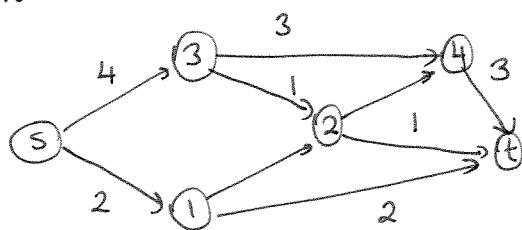
Augmenting path:

$$s \rightarrow 1 \rightarrow t$$

$$s = 1$$

Augment:

e.

flow = 6  $\rightarrow$  optimal

since we have a cut  $\{s\}, \{1, 2, 3, 4, t\}$   
with value 6  
or equivalently  $\exists$  no augmenting path  
from  $s$  to  $t$ .

Note: Try to write the dual problem and observe that this cut gives dual variables which are feasible. So we have optimality.