

Duality: (LP)

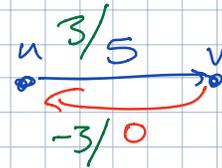
• Flow-cut duality [Ford Fulkerson]

• cut-cycle duality [Whitney]

min  $\rightarrow$  shortest  $\downarrow$   
shortest path

To compute max flow fast, combine  
flows in  $G \Leftrightarrow$  shortest paths in  $G^*$

(Directed) graph  $G = (V, E)$  Darts  
 capacity  $c: D \rightarrow \mathbb{R}_{\geq 0}$

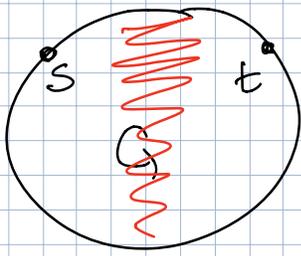


Flow  $\phi: D \rightarrow \mathbb{R}$  s.t.  $\phi(d) = -\phi(\text{rev}(d))$   
*antisymmetric*

balance:  $\sum_u \phi(u \rightarrow v) = 0$  for all  $v \neq s, t$

value:  $\sum_v \phi(s \rightarrow v) = \sum_v \phi(v \rightarrow t)$

feasible:  $\phi(u \rightarrow v) \leq c(u \rightarrow v)$

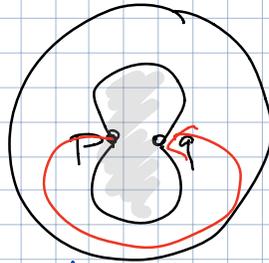
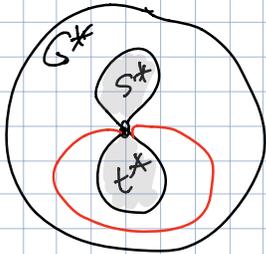


$(s,t)$ -planar graph

[Hassin'71]

terminals  $s, t$  on outer face

$\min(s,t)$  cut "is" shortest path from  $p$  to  $q$



For any vertex  $x$  in  $G^*$

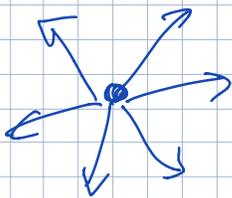
$\text{dist}(x) = \text{sh. path distance from } p \text{ to } x$

$f(u \rightarrow v) = \text{dist}(\text{right}(u \rightarrow v)) - \text{dist}(\text{left}(u \rightarrow v))$

Dual:  $f(x \rightarrow y) = \text{dist}(y) - \text{dist}(x) \leq c(x \rightarrow y)$

$$\boxed{\text{dist}(y) \leq \text{dist}(x) + c(x \rightarrow y)}$$

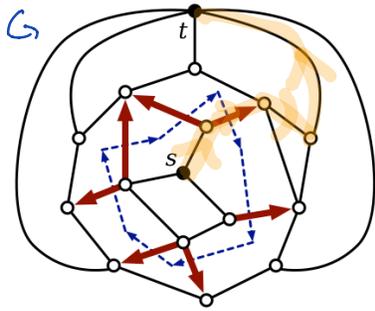
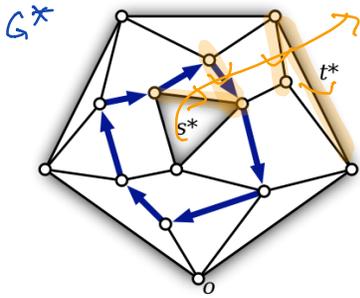
so  $f(e) \leq c(e)$  for all primal edges



$\sum_w f(v \rightarrow w) = 0$  so it's a flow!  
because dists cancel!

Dijkstra  $\rightarrow O(n \log n)$

Fast  $\rightarrow O(n)$



[Venkatesan 83]

Given  $G$  and  $\lambda$

compute a feasible flow with value  $\lambda$   
in  $O(n \log n)$  time  
or report none exists

Compute dist in  $G_\lambda^*$   
if succeed dist  $\lambda \rightarrow$  Flow value  $\lambda$   
if fail  $\lambda >$  max flow value  
 $\rightarrow$  cut with cap  $<$   $\lambda$

$P =$  any path  $s \rightarrow t$  in  $G$

$$\pi(d) = \begin{cases} 1 & \text{if } d \in P \\ -1 & \text{if } \text{rev}(d) \in P \\ 0 & \text{o/w} \end{cases}$$

For any cycle/closed walk  $w$  in  $G^*$

$$\sum_{e \in w} \pi(e^*) = \text{wind}(w, s^*) = \pi(w)$$

$$\text{for any path } \alpha \text{ in } G^* \quad \pi(\alpha) = \sum_{e \in \alpha} \pi(e)$$

Define  $c_\lambda(u \rightarrow v) = c(u \rightarrow v) - \lambda \cdot \pi(u \rightarrow v)$

$\lambda > 0$

Define  $\text{dist}_\lambda$  in  $G^*$  wrt cost  $c_\lambda$

$G_\lambda^*$  has neg. cycle  $\iff$  No feasible flow in  $G$  with value  $\lambda$

$\implies$

Let  $C$  be a neg cycle in  $G^*$

$$0 > \sum_{p \rightarrow q \in C} c_\lambda(p \rightarrow q) = \sum_{p \rightarrow q} c(p \rightarrow q) - \lambda \cdot \sum_{p \rightarrow q} \pi(p \rightarrow q)$$

$$= c(C) - \lambda \cdot \pi(C)$$

$$\text{so } \pi(C) > 0 \implies \pi(C) = 1$$

Then  $C^*$  is an  $(s, t)$  cut with  $c(C^*) < \lambda$   $\square$

$\Leftarrow$  Suppose  $\text{dist}_\lambda$  is well-defined

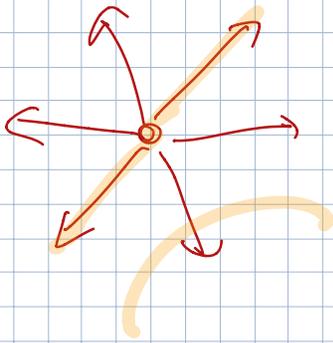
$$\text{slack}_\lambda(p \rightarrow q) = \text{dist}_\lambda(p) + c_\lambda(p \rightarrow q) - \text{dist}_\lambda(q) \geq 0$$

for all  $u \rightarrow v$   
in  $G^*$

$$\boxed{\Phi_\lambda(p \rightarrow q) = \text{dist}_\lambda(q) - \text{dist}_\lambda(p) + \lambda \cdot \pi(p \rightarrow q)}$$

$$= c(p \rightarrow q) - \text{slack}_\lambda(p \rightarrow q)$$

$$\leq c(p \rightarrow q)$$



$$\sum_v \Phi_\lambda(u \rightarrow v) = \sum_v \lambda \cdot \pi(u \rightarrow v) = 0$$

except  $v = s$   
or  $v = t$

$\Phi$  is a feasible flow  
with value  $\lambda$ !  $\square$

We can detect neg cycles  $O(n \log^2 n)$  time  
Binary search over  $\lambda$

maxflow in  $O(n \log^2 n \log n U)$  time

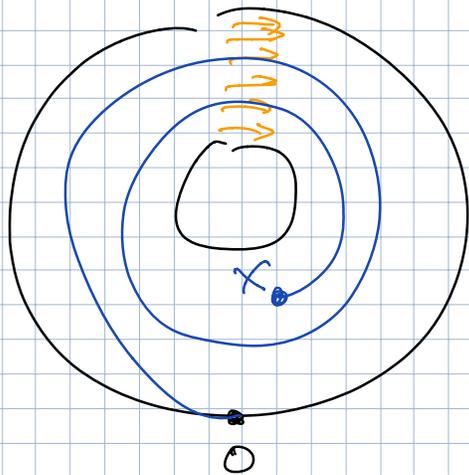
$U = \max \text{cap}$

(all caps are ints)

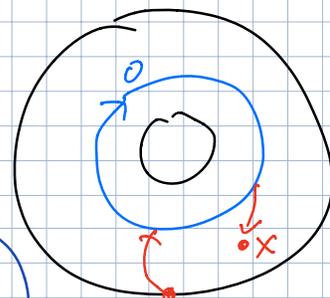
# Parametric shortest paths

$$C_\lambda(u \rightarrow v) = c(u \rightarrow v) - \lambda \cdot \pi(u \rightarrow v)$$

vary  $\lambda$  continuously, maintain  $\text{dist}_\lambda \rightarrow \phi_\lambda$

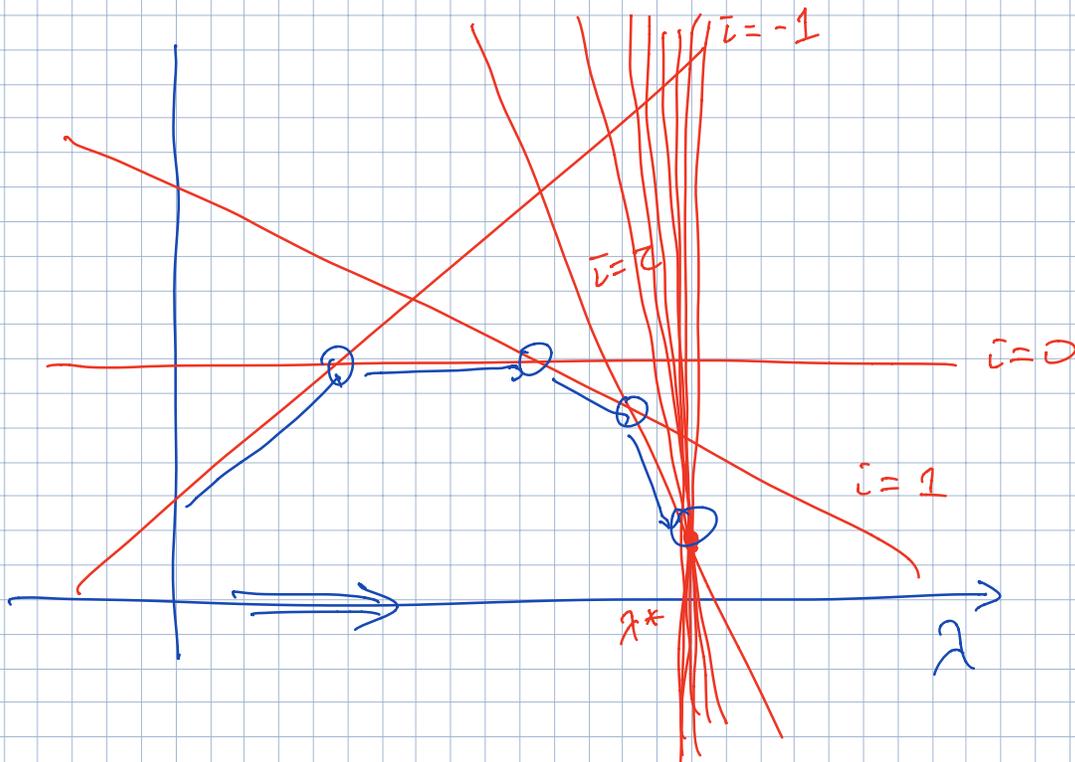


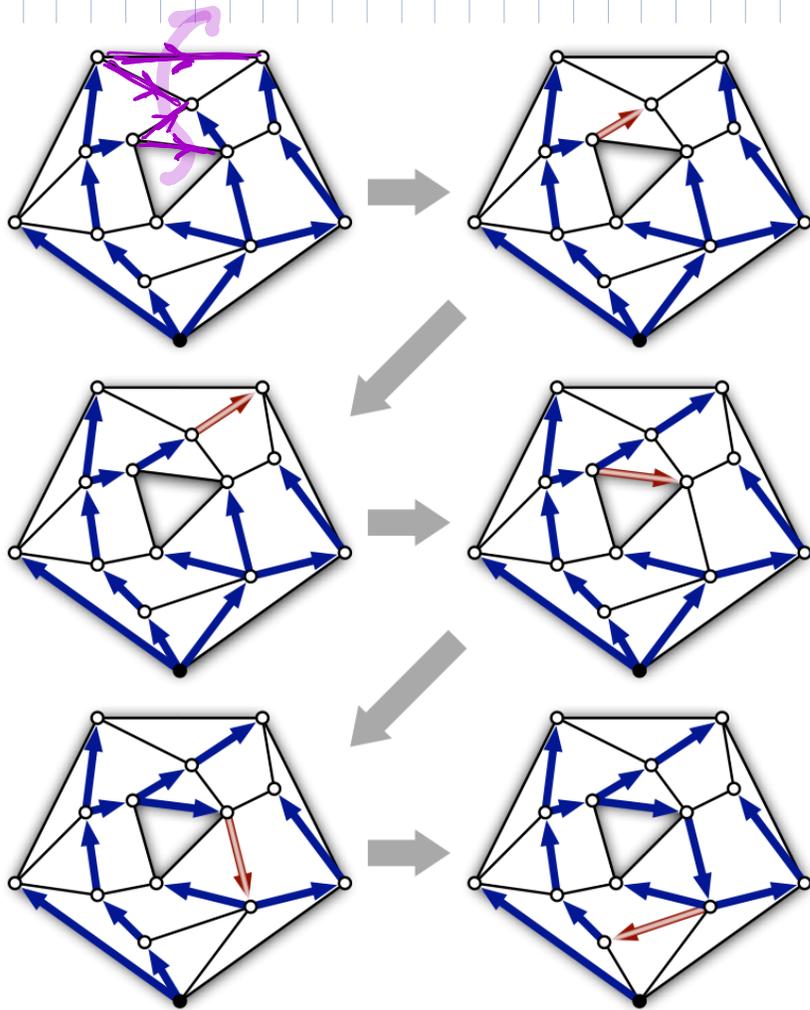
$\text{path}_i(x)$  = shortest "path"  
from  $o$  to  $x$  with  
 $\pi = i$



$$\text{dist}_\lambda(x) = \min_i (C_\lambda(\text{path}_i(x)))$$

$$C_\lambda(\alpha) = c(\alpha) - \lambda \cdot \pi(\alpha)$$

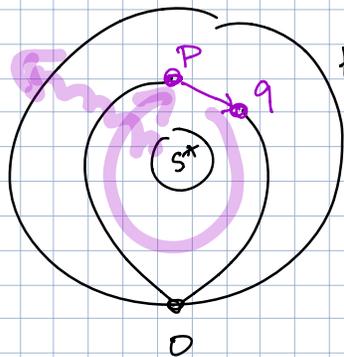




① Which edges can pivot next?  
 Which edges have decreasing slack?

$$\begin{aligned} \text{slack}_\lambda(p \rightarrow q) &= \text{dist}_\lambda(p) + c_\lambda(p \rightarrow q) - \text{dist}_\lambda(q) \\ &= \text{slack}_0(p \rightarrow q) - \lambda \cdot \text{slack}'(p \rightarrow q) \end{aligned}$$

- If  $p \rightarrow q$  is in  $T_\lambda$ ,  $\text{slack} \equiv 0 \Rightarrow \text{slack}' = 0$



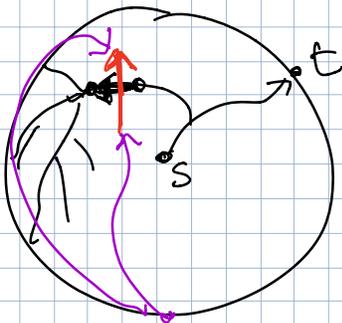
$$\begin{aligned} \text{slack}'(p \rightarrow q) &= \pi(\text{cycle}(T_\lambda, p \rightarrow q)) \\ &\uparrow \\ &\text{path}_\lambda(p) \cdot p \rightarrow q \cdot \text{rev}(\text{path}_\lambda(q)) \end{aligned}$$

so  $\text{slack}(p \rightarrow q)$  is decreasing iff  $\pi(\underbrace{\text{cycle}(T_\lambda, p \rightarrow q)}_C) = 1$

$\Leftrightarrow C^*$  is an (s,t)-cut

Tree-cotree decomposition

$(L_\lambda, T_\lambda)$  —  $T_\lambda$  is shortest-path tree in  $G_\lambda^*$   
 $L_\lambda$  is set of loose edges  $uv$  in  $G$   
 $\text{slack}(u \rightarrow v) > 0$   
 $\text{slack}(v \rightarrow u) > 0$



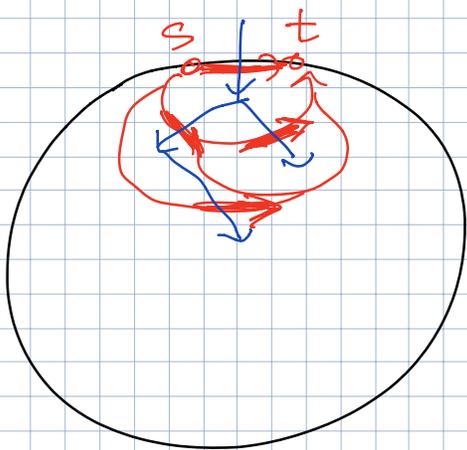
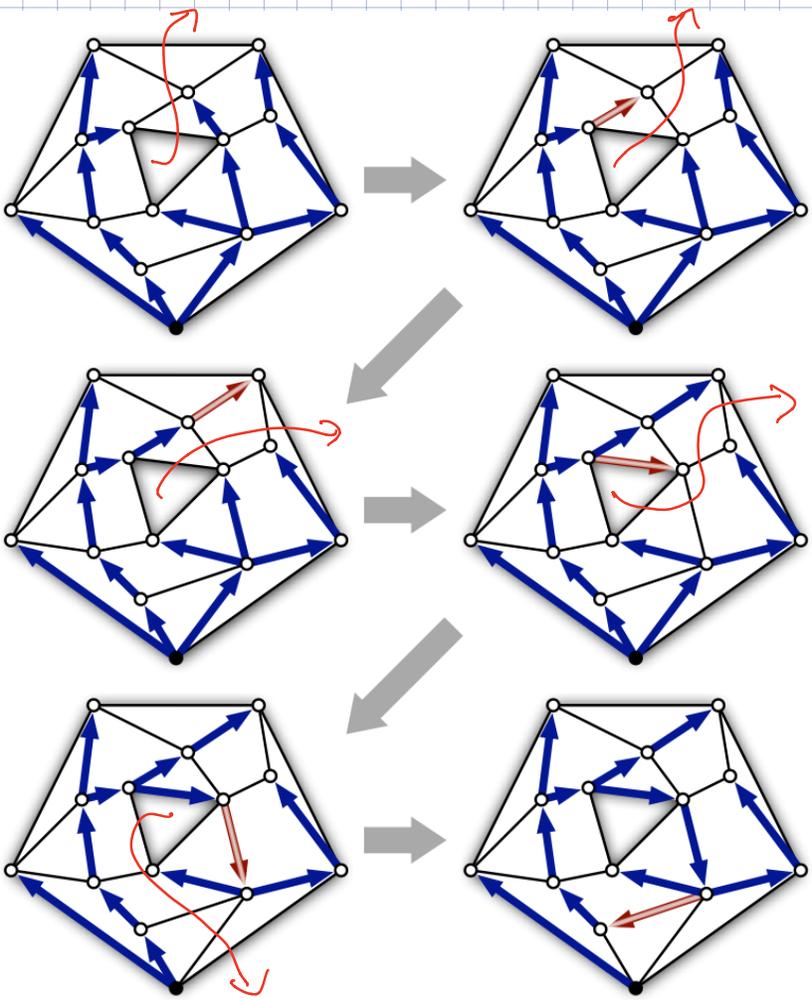
$LP_\lambda =$  unique path in  $L_\lambda$  from  $s$  to  $t$   
Claim:  $\text{slack}(p \rightarrow q)$  is decreasing  
 $\Leftrightarrow (p \rightarrow q)^* \in LP_\lambda$

$\rightarrow \phi(p \rightarrow q)$  is increasing

This is Ford-Fulkerson

Borradale Klein — "leftmost augmenting path"

CF FF — "uppermost aug. path"  
 $=$  Dijkstra



PLANARMAXFLOW( $G, c, s, t$ ):

Initialize the spanning tree  $L$ , predecessors, and slacks

$\lambda \leftarrow 0$

while  $s$  and  $t$  are in the same component of  $L$

$LP \leftarrow$  the path in  $L$  from  $s$  to  $t$

$p \rightarrow q \leftarrow$  the edge in  $LP^*$  with minimum slack

$\Delta \leftarrow \text{slack}(p \rightarrow q)$      $\lambda \leftarrow \lambda + \Delta$

for every edge  $e$  in  $LP$

$\text{slack}(e^*) \leftarrow \text{slack}(e^*) - \Delta$

$\text{slack}(\text{rev}(e^*)) \leftarrow \text{slack}(\text{rev}(e^*)) + \Delta$

delete  $(p \rightarrow q)^*$  from  $L$

if  $q \neq o$  ((that is, if  $\text{pred}(q) \neq \emptyset$ ))

insert  $(\text{pred}(q) \rightarrow q)^*$  into  $L$

$\text{pred}(q) \leftarrow p$

for each edge  $e$

$\phi(e) \leftarrow c(e) - \text{slack}(e^*)$

return  $\phi$

$\leftarrow$  compute  $T_0$

Data structures:

Store  $L$  in dynamic tree structure

$O(\log n)$  am. time per operation

Overall time:

$O(n \log n + N \log n)$

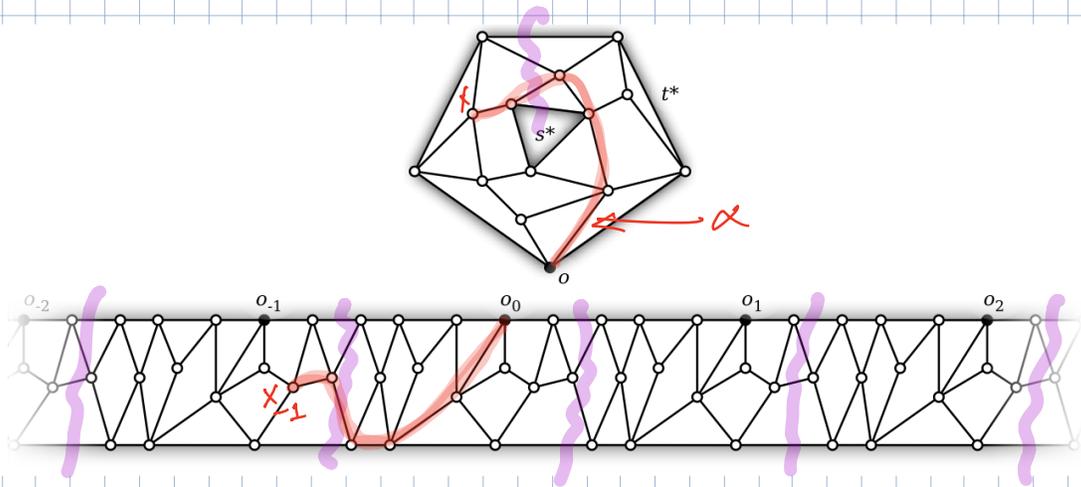
where  $N = \# \text{pivots}$

Figure 4. Our planar maximum flow algorithm.

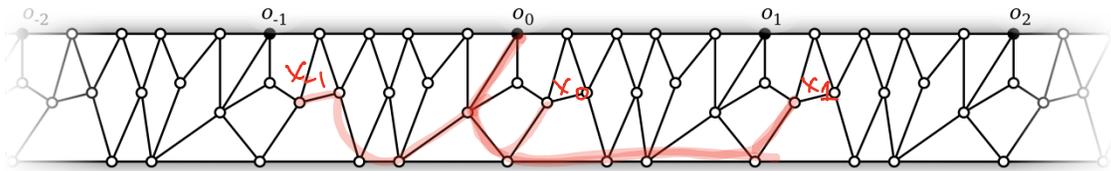
Network Simplex

Claim:  $N = O(n)$

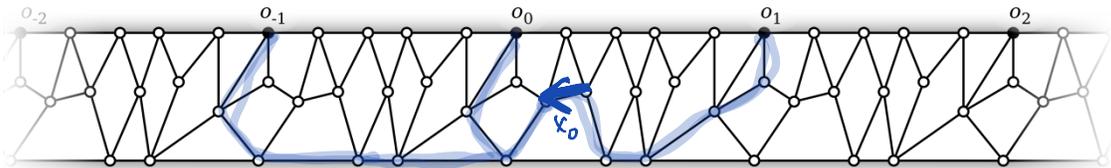
Each dart  $p \rightarrow q$  pivots into  $T_1$  at most once



path  $\alpha$  from  $o$  to  $x$  lifts  
 $\hat{\alpha}$  from  $o_0$  to  $x_{\pi(x)}$   
 path  $i(x)$  "is" shortest path from  $o_0$  to  $x_i$   
 to  $x_0$  from  $o_{-i}$



$\gamma_0 \rightarrow x_0$



Tree-disk lemma

