

Maximum Flows in Planar Graphs

We've seen two kinds of duality

- Flow \leftrightarrow cut duality [Ford-Fulkerson]
- Cut \leftrightarrow cycle duality [Whitney]
- Min cut \leftrightarrow shortest cycle [Itai-Shiloach]
- ↓
- shortest paths [Zemf, et al]

Antisymmetric Flows

$$\phi: D \rightarrow \mathbb{R} \text{ is Flow}$$

$$\phi(d) = -\phi(\text{rev}(d))$$

$$\phi(d) \leq c(d)$$

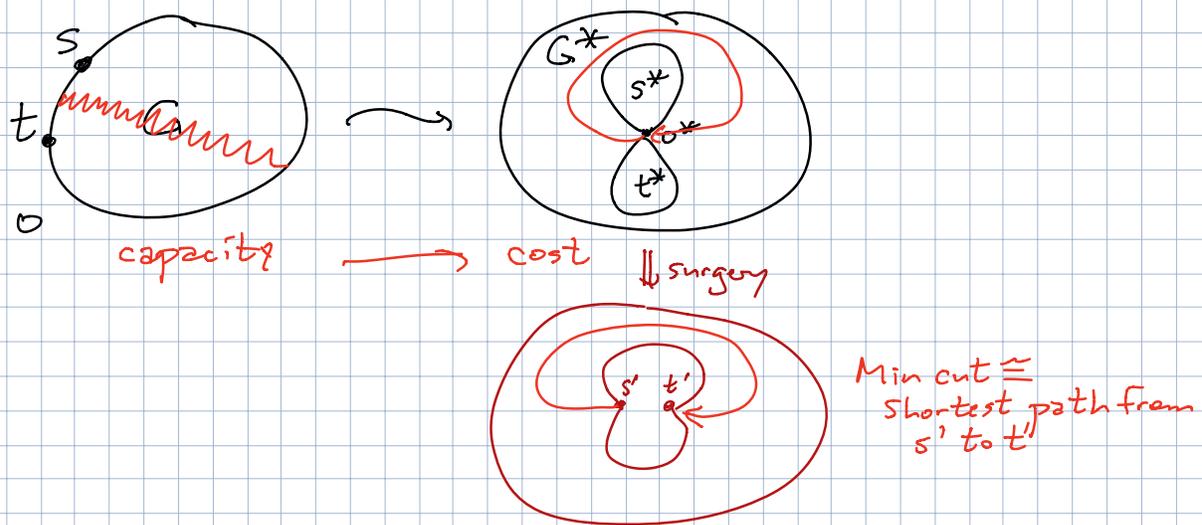
capacity

$$\sum_v \phi(u \rightarrow v) = 0 \quad \text{if } u \neq s, t$$

conservation

Combining these reveals a duality between Flows in G and shortest paths in G^*

[Hassin '81] Suppose s and t are on same (outer) face



But we can actually recover the max Flow (function, not just value)

For every dual vertex f^* , define $\text{dist}(f^*) = \text{distance from } s^*$

for each edge $u \rightarrow v$ in G define

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

$$= \text{dist}(y) - \text{dist}(x)$$

where $(u \rightarrow v)^* = (x \rightarrow y)$

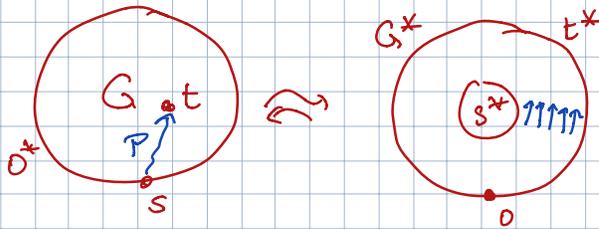
For every dual edge $x \rightarrow y$: • $\text{dist}(x) + c(x \rightarrow y) \geq \text{dist}(y)$
 $\Rightarrow \text{dist}(y) - \text{dist}(x) \leq c(x \rightarrow y)$

$$\Rightarrow F(u \rightarrow v) \leq c(u \rightarrow v)$$

conservation: dist terms cancel.

But why is $F(u \rightarrow) \geq 0$?! ← Reversed edges are feasible, too

[Venkatesan '87] Finding a feasible flow with value λ .



Fix arbitrary path P from s to t .

Define residual capacity

$$c_\lambda(u \rightarrow v) = \begin{cases} c(u \rightarrow v) - \lambda & \text{if } e \in P \\ c(u \rightarrow v) + \lambda & \text{if } e \in \text{rev}(P) \\ c(u \rightarrow v) & \text{otherwise} \end{cases}$$

Residual capacities could be negative — It's okay; relax!

Use c_λ as cost function in G^* → dual residual network G_λ^*

$\text{dist}_\lambda^*(f^*) =$ distance from o to f^* in G^* wrt c_λ

well-defined iff no negative cycles in G_λ^*

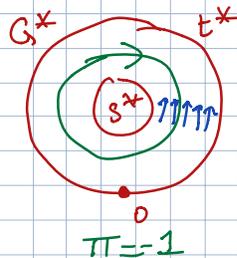
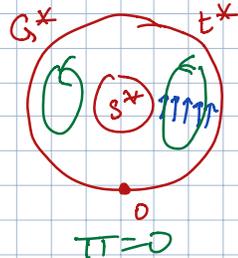
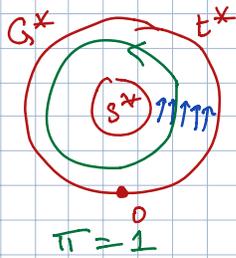
• Suppose neg. cycle C in G_λ^*

$$c_\lambda(C) = \sum_{e \in C} (c(e) - \lambda \pi(e)) < 0$$

$$\text{but } \sum c(e) \geq 0$$

$$\lambda \geq 0$$

$$\text{so } \pi(C) \geq 0$$



$\pi(C) =$ winding # of C around s^*

Let C be simple neg cycle

→ $\pi(C) = 1$

Then C^* is an (s, t) -cut with

$$\sum_{e \in C^*} c(e) > \lambda \quad \text{so } \text{mincut} = \text{maxflow} < \lambda.$$

• Suppose dist_λ well-defined

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

$$\text{Define } \phi_\lambda(p \rightarrow q) = \text{dist}_\lambda(q) - \text{dist}_\lambda(p) + \lambda \cdot \pi(p \rightarrow q) \\ = c(p \rightarrow q) - \text{slack}_\lambda(p \rightarrow q)$$

$$\text{So } \phi_\lambda(p \rightarrow q) \leq c(p \rightarrow q)$$

$$\text{Conservation: } \sum_u \phi(u \rightarrow v) = \sum_u \lambda \pi(u \rightarrow v) = \begin{cases} \lambda & \text{if } v = s \\ -\lambda & \text{if } v = t \\ 0 & \text{o/w} \end{cases}$$

dist/cancel

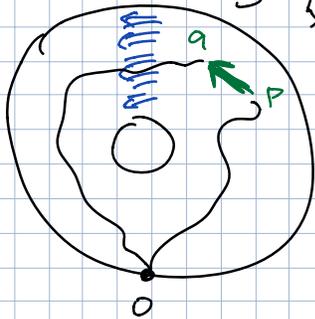
So ϕ_λ is a valid flow with value λ .

Parametric — Vary λ continuously from 0 until shortest paths break

dist_λ and ϕ_λ vary continuously, but

shortest path tree T_λ changes by discrete pivots, just like in MSSP algorithm.

Edge $p \rightarrow q$ pivots into T_λ just before $\text{slack}_\lambda(p \rightarrow q) < 0$
 $x \rightarrow q$ pivots out



while T_λ is fixed

$$\text{slack}(\lambda, p \rightarrow q) = \text{slack}_0(p \rightarrow q) - \lambda \cdot \text{slack}'(p \rightarrow q)$$

$$\text{slack}'(p \rightarrow q) = \pi(\text{cycle}(T_\lambda, p \rightarrow q))$$

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

$$\text{slack}'(p \rightarrow q) = \text{dist}'(p) + \pi(p \rightarrow q) - \text{dist}'(q)$$

$$= \pi(\text{path}_\lambda(p)) + \pi(p \rightarrow q) + \pi(\text{rev}(\text{path}_\lambda(q)))$$

$$= \boxed{\pi(\text{cycle}(T_\lambda, p \rightarrow q))} \in \{-1, 0, 1\}$$

because cycle is simple.

$p \rightarrow q$ is active if $\text{slack}'(p \rightarrow q) = 1$

Let $L_\lambda =$ complementary spanning tree of G

↳ loose edges: $\text{slack} > 0$

$LP_\lambda =$ unique path from s to t in L_λ

Lemma: e^* is active iff $e \in LP_\lambda$

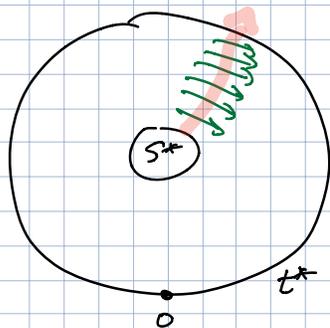
Proof: Edges of T_λ have $\text{slack} = 0$ so $\text{slack}' = 0$ so inactive

e^* active $\Leftrightarrow \pi(\underbrace{\text{cycle}(T_\lambda, e^*)}) = 1 \Leftrightarrow C(e)$ is an (s, t) -cut
 ↳ $(\text{cycle}(T_\lambda, e^*))^*$

$\Rightarrow LP_\lambda$ contains at least one edge of $C(e)$

But e is only loose edge of $C(e)$, so $e \in LP_\lambda$

OTOH, if $e \in LP_\lambda$, then $C(e)$ is an (s, t) -cut. \square



So as we increase λ

$\text{slack}_\lambda(e)$ decreases for all $e \in LP_\lambda$

increases for all $e \in \text{rev}(LP_\lambda)$

Equivalently, $\phi_\lambda(e)$ increases along LP_λ
 decreases along $\text{rev}(LP_\lambda)$

Increasing λ pushes flow from s to t along LP_λ

We pivot e into T_λ when $\text{slack}(e) \searrow 0$

when $\phi(e) = c(e)$
 when e is saturated!

When we pivot $p \rightarrow q$ into T_λ , we pivot $\text{pred}(q) \rightarrow q$ out
 out of L_λ into L_λ

We get a new augmenting path LP_λ !

Eventually, the pivot creates a cycle C^* in T_λ
 disconnects L_λ

$C_\lambda(C^*) = 0 \Leftrightarrow \phi(C) = c(C) \Leftrightarrow \phi$ is max flow, C is min cut!
 ↳ no more augmenting paths!

Implementation:

Compute initial ^{dual} shortest path tree T_0
 Maintain primal spanning tree L
 While L is connected
 $LP \leftarrow$ path from s to t in L \otimes
 $p \rightarrow q \leftarrow$ min slack edge in LP^* \otimes
 Push $slack(p \rightarrow q)$ through LP \otimes
 delete $(p \rightarrow q)^*$ from L \otimes
 insert $(pred(q) \rightarrow q)^*$ into L \otimes
 $pred(q) \leftarrow p$
 $\phi = c - slack$

Don't need to maintain T_λ
 Don't need to maintain λ
 Don't need to compute $P!$

Dynamic forest needs $O(\log n)$ time for each operation \otimes

[Borradale Klein]: Leftmost augmenting path

Analysis:

Recall $dist_\lambda(q) = \min_i (c_\lambda(path_i(q)))$

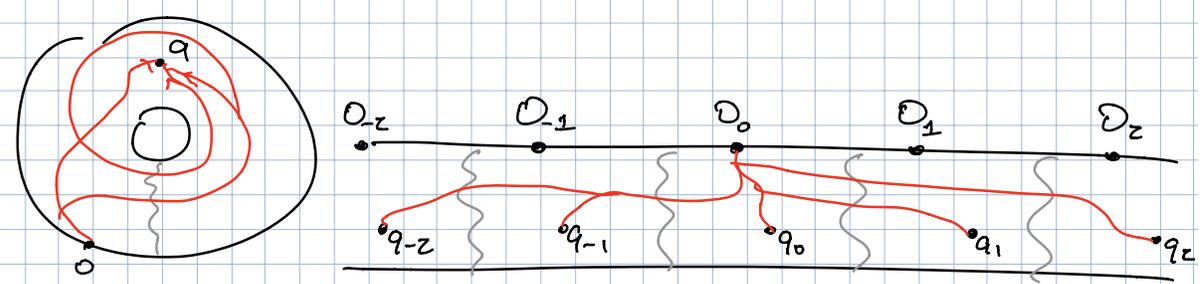
Shortest ^{walk} path from o to q with $\pi = i$

Whenever $path_\lambda(q)$ changes

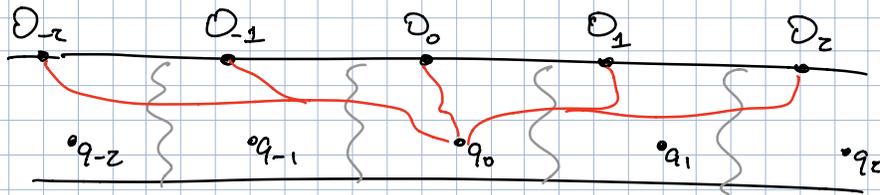
$\pi(path_\lambda(q))$ increases by 1 (by $\pi(cycle(T, p \rightarrow q))$)

Question:

For which i is $p \rightarrow q$ in $path_i(q)$?



All T_λ are subtrees of shortest path tree @ o_0 in \hat{G}



Disk-tree lemma implies $p \rightarrow q \in T_\lambda$ for a
contiguous range of λ

$\Rightarrow p \rightarrow q$ pivots into T_λ at most ONCE

$\Rightarrow O(n)$ pivots!

$\Rightarrow O(n \log n)$ time!

