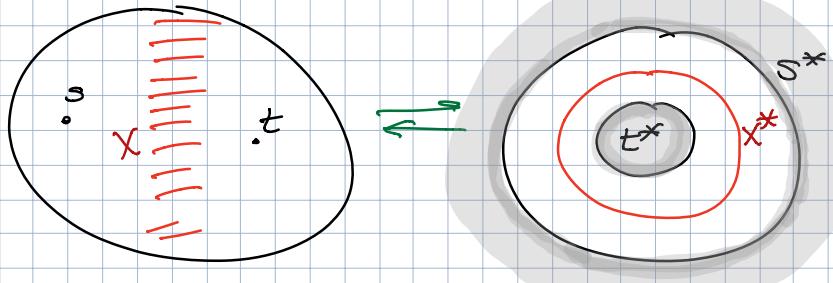


## Minimum Cuts in Planar Graphs



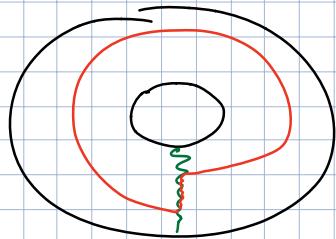
Whitney duality: minimal cut  $\Leftrightarrow$  simple cycle  
 separating  $s, t$       separating  $s^*$  and  $t^*$   
 $\min. wt$        $\min. nt$

So minimum  $(s, t)$ -cut in  $G$  is dual to  
 minimum generating cycle in annulus  $G^* \setminus (s^* \cup t^*)$

$\hookrightarrow$  = homotopic to boundary

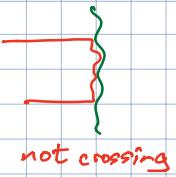
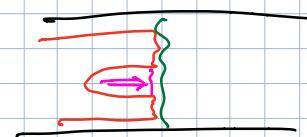
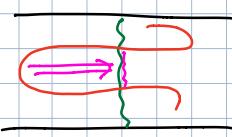
= winding # 1

[= reduced crossing seq length 1]



Let  $\pi$  = shortest path from  $s^*$  to  $t^*$

Shortest gen. cycle  $\lambda$  crosses  $\pi$  exactly once  
 $\lambda \cap \pi$  is an interval



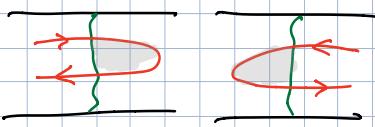
Consider crossing sequence

$+$  =  $\longrightarrow$   
 $-$  =  $\longleftarrow$

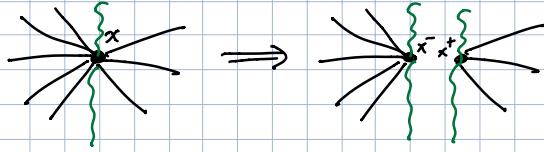
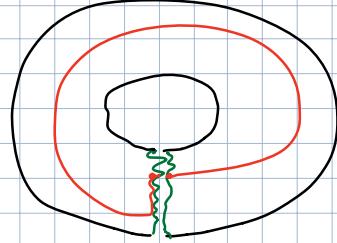
Boundary has crossing sequence +  
 So any cycle hom. to boundary  
 has reduced x-seq +

So  $|X| \geq 1$ , and

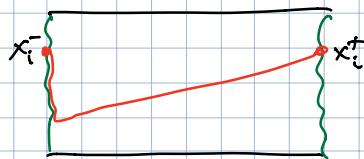
If  $|X| > 1$ , must contain  $+-$  or  $-+$ , so we can shortcut



Cut open along  $\pi$ : Duplicate verts+edges, split incident edges



Now looking for shortest path  
between matching vertices  
on  $\pi^-$  and  $\pi^+$



Suppose  $\pi$  has  $k$  edges.

Naïve:  $k \times$  Dijkstra —  $O(kn \log n)$  time

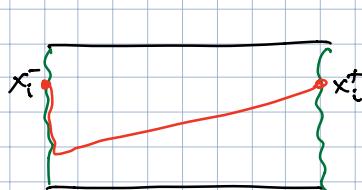
$k \times$  Fast SP —  $O(kn)$

MSSP —  $O(n \log n)$

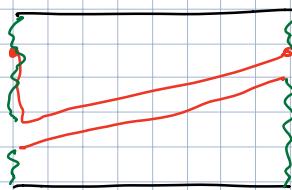
[Reif] Divide+conquer —  $O(n \log k)$

Split along median path from  $x_{k/2}^-$  to  $x_{k/2}^+$

Recurse above + below



Let  $n = \# \text{verts of } G$   
 $= \# \text{faces of } G^*$   
Euler:  $= \partial(\# \text{verts of } G^*)$



$$\begin{aligned} T(n, k) &= O(n \cancel{\log n}) + T(n_1, k/2) + T(n_2, k/2) \\ &= O(n \cancel{\log n} \log k) \end{aligned}$$

FIR-Dijkstra — Shortest paths in dense distance graphs

Recall def. DDG:

- Build nice  $r$ -division:  $\Theta(\frac{n}{r})$  pieces, each with  $O(r)$  verts  
 $O(\sqrt{r})$  bdry verts  
disk with  $O(1)$  holes

- Replace each piece with clique over bdry verts

→ Distances via MSSP in  $O(r \log r)$  per piece  
 $= O(n \log r)$  total

$- O(n/r)$  vertices  
 $- O(n)$  edges

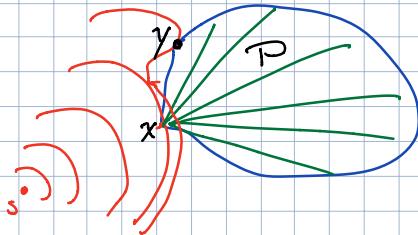
So Dijkstra runs in  $O(n + \frac{1}{r} \log n)$  time

$O(n)$   
distances  
or  $O(\log r)$  per  
distance later

We want to lose the  $n/r$  term — Don't look at every edge

FIR exploit Monge structure!

For now, assume pieces are disks



Let  $P$  be any piece  
when Dijkstra wavefront hits  $P$  at vertex  $x$   
relax all edges  $x \rightarrow z$  in  $P$

Then later when wave hits  $y$ ,  
relax all edges  $y \rightarrow z$

Define  $M[v, u] = \text{dist}(v) + w(v \rightarrow u)$

We discover  $\text{dist}(v)$  in increasing order

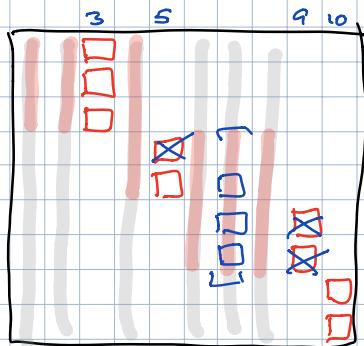
→ Columns of  $M$  "revealed" one at a time

Ultimately we need  $\min_u (\text{dist}(u) + w(u \rightarrow v))$  for all  $v$

→ Need minima of every row of  $M$ .

Monge heap: Maintain row minima of an  $n \times n$  Monge array  
as columns revealed

Trick: Row minima are monotone



Suppose columns  $j_1 < j_2 < \dots < j_k$  revealed so far

THIS IS  
INCOMPLETE

Each col. contains minima in an interval of rows  $L[j] \dots U[j]$

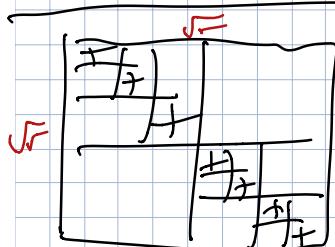
If unseen column  $j_i < j < j_{i+1}$

$$L[j_i] \leq L[j] \leq L[j_{i+1}]$$

$$U[j_i] \leq U[j] \leq U[j_{i+1}]$$

$$L[3]=4, U[5]=5$$

So when new column is revealed, binary search for  $L[i]$  and  $U[i]$  in  $O(\log n)$  time



Distance arrays decompose into  $O(r)$  monge arrays

Each vertex appears in  $O(\log r)$  monge arrays

Back to FR-Dijkstra:

Keep bdy verts in global priority queue

Pull min. vertex, relax outgoing edges

↳ reveal columns in  $O(\log r)$  Monge arrays  
for each incident piece

Keep a global heap of all  $O(\sqrt{r})$  vertices

↳ Actually a heap of  $O(\frac{n}{\sqrt{r}})$  Monge arrays,  
each maintaining its own minimum row

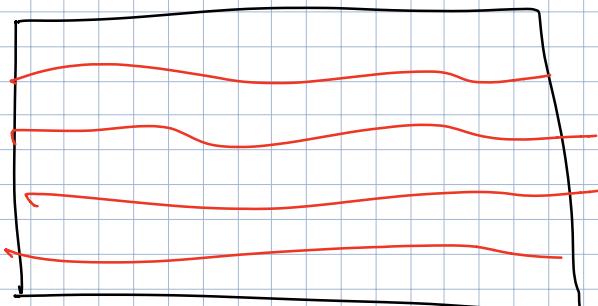
Each vertex in  $O(\log r)$  arrays, so ExtractMin takes  
 $O(\log \log r)$  time.

Within each piece:

$$\begin{aligned} \text{Total Time for first column: } T_F(h) &= O(h) + ZT_F(h/c) \\ &\Rightarrow O(\sqrt{r} \log r) \end{aligned}$$

$$\begin{aligned} \text{Total Time for all other columns: } T_O(h) &= O(h \log h) + ZT_O(m/c) \\ &\Rightarrow O(\sqrt{r} \log^2 r) \end{aligned}$$

Each relaxation could change  $O(\sqrt{r})$  distances



Find  $\frac{n}{\log n}$  paths Reif

$$\begin{aligned} O(n) + \\ O\left(\frac{n}{\sqrt{r}} \log^2 n\right) + 2T(n, r) \\ = O\left(\frac{1}{\sqrt{r}} \log^3 n\right) + O(n \log n) \end{aligned}$$

Recurse in both sides:

$$T(n) = O(n) + O(\log n) \cdot T\left(\frac{n}{\log n}\right) = O(n \log \log n)$$