Routing Air Traffic Flows: from Continuous to Discrete and Back

Valentin Polishchuk

Helsinki Institute for Information Technology

joint work with

Joondong Kim
Joseph Mitchell

Stony Brook University

Jimmy Krozel
Joseph Prete

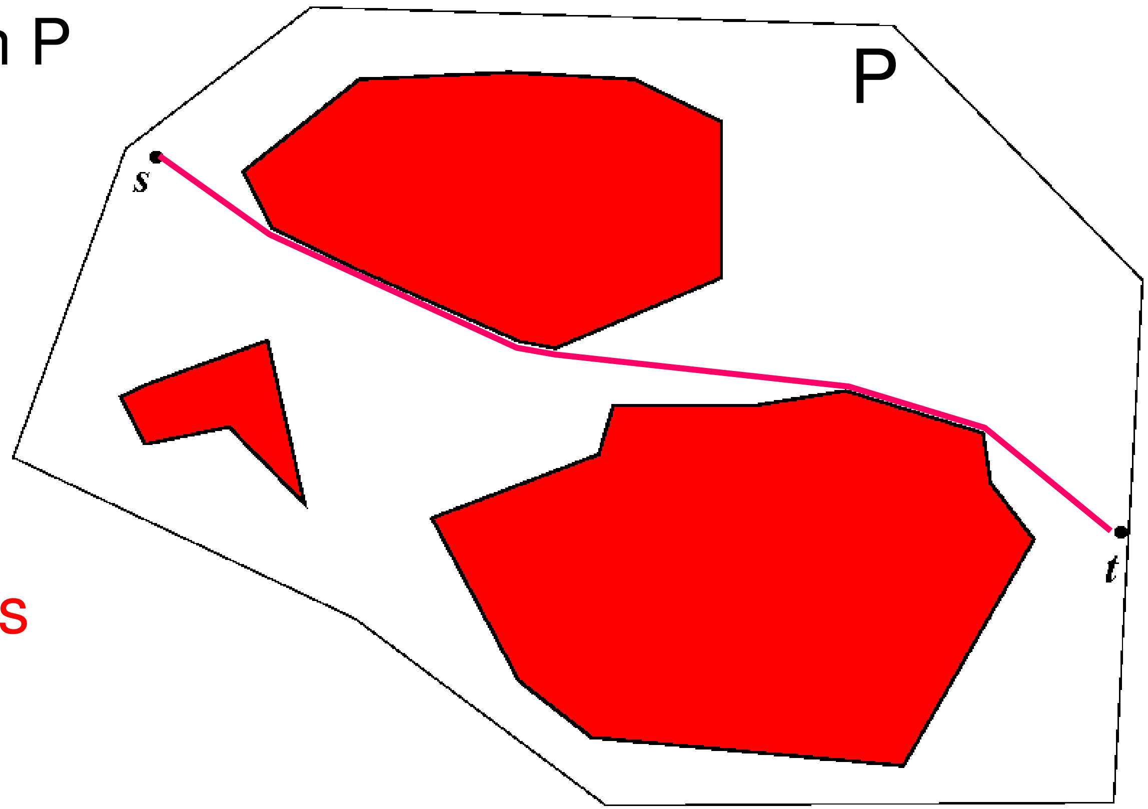
Metron Aviation

Arto Vihavainen

University of Helsinki

Geometric Shortest Paths

- Given:
  - Polygonal domian P
    - obstacles (holes)
  - Points s and t

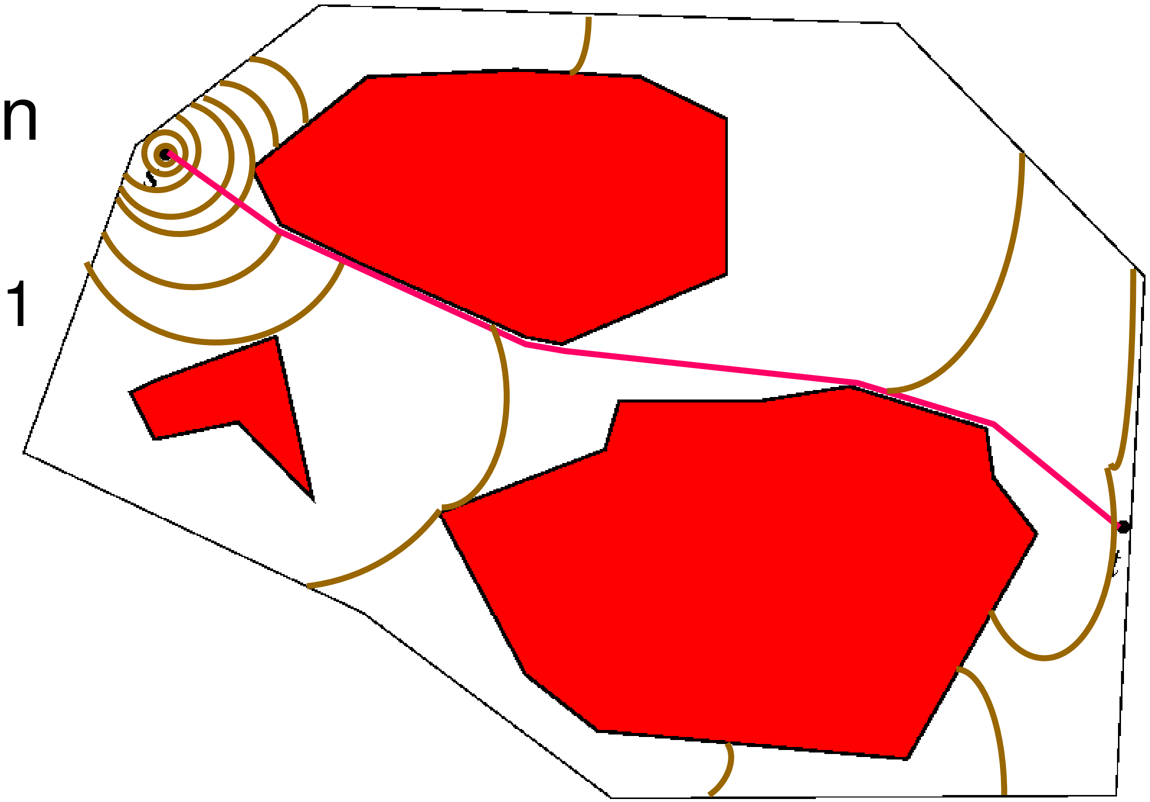


- Find:
  - shortest s-t path
  - avoiding obstacles

Continuous Dijkstra

[Mitchell'86]

- Wave propagation
  - starts from s
  - travels at speed 1
- Wavefront at τ
  - bd of what's reached by τ



| SP from s to t | =

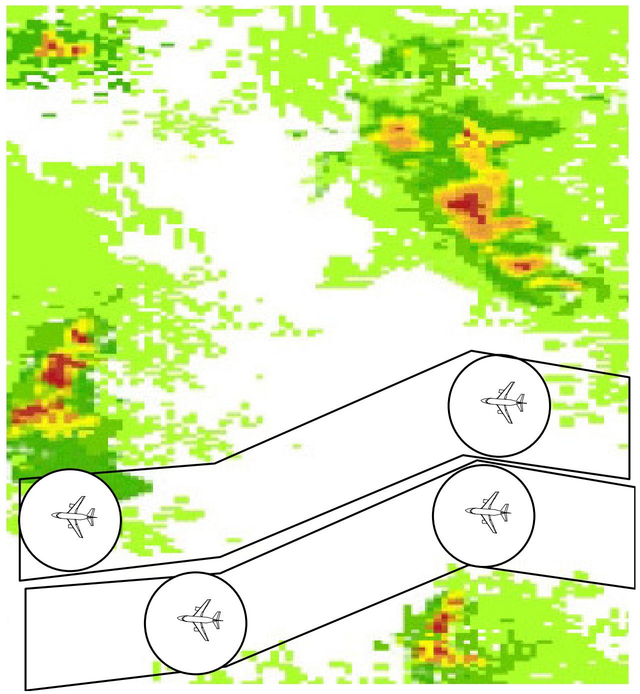
time when wavefront hits t

Applications

- VLSI
- Robotics
- Sensor networks
- Air Traffic Management
  - **safety margin**

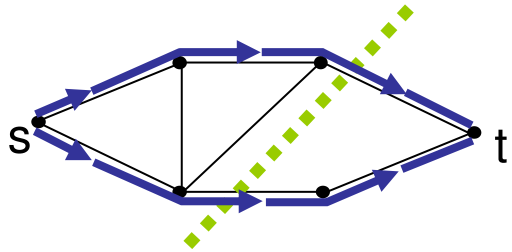
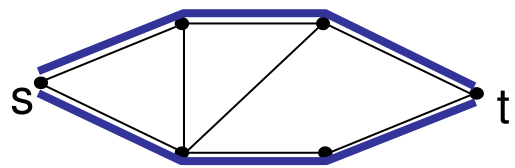
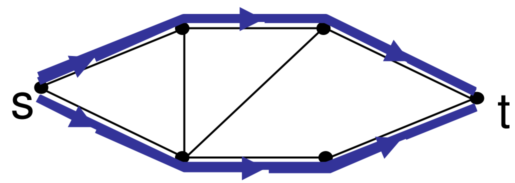
Multiple disjoint "**thick**" paths

How to find thick paths in a polygonal domain?



Disjoint Paths in Graphs

Related to Network Flows

MaxFlow/MinCut Theorem		maxflow = mincut 2 = 2
Menger's Theorem		max # of disjoint s-t paths = min # of vertices to disconnect s and t 2 = 2
Flow Decomposition Theorem		flow = union of paths

Flows and Paths
in
Geometric Domains

MaxFlow: Problem Statement

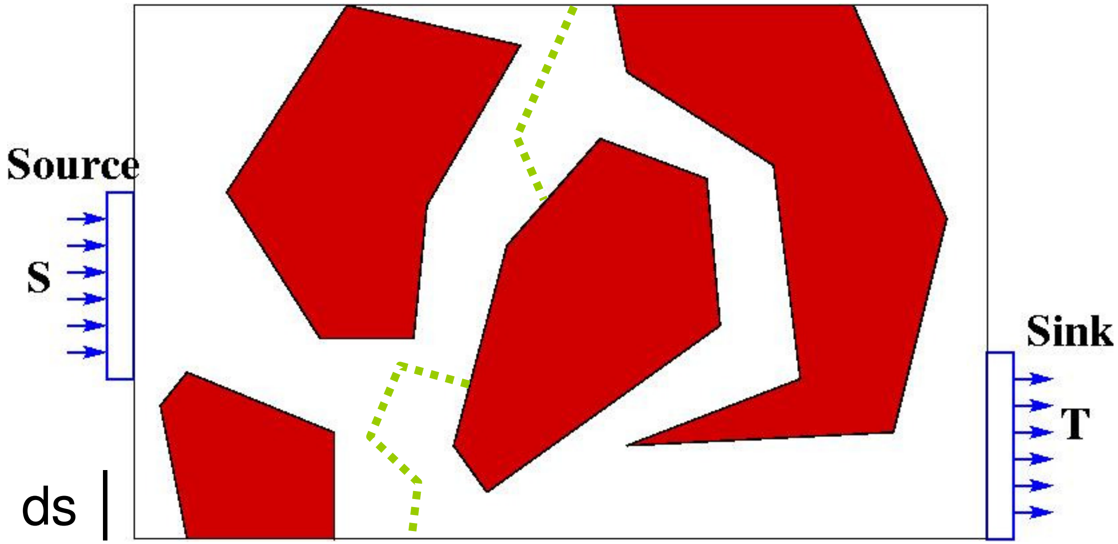
Given: Polygonal domain P

with holes

source and sink **S** and **T**

Flow

vector field $\sigma: P \rightarrow R^2$



div $\sigma = 0$ inside P

$\sigma \bullet n = 0$ on $\partial P \backslash \{S,T\}$

$|\sigma| \leq 1$ – capacity

$$V = \left| \int_S \sigma \bullet n \, ds \right| = \left| \int_T \sigma \bullet n \, ds \right|$$

Cut: Partition P

S in one part, **T** in the other

Capacity: Length of bd between parts

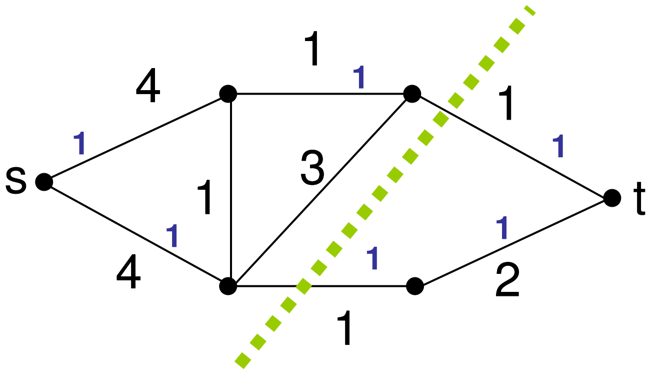
counted within P (not within holes)

MaxFlow

Find σ that maximizes V

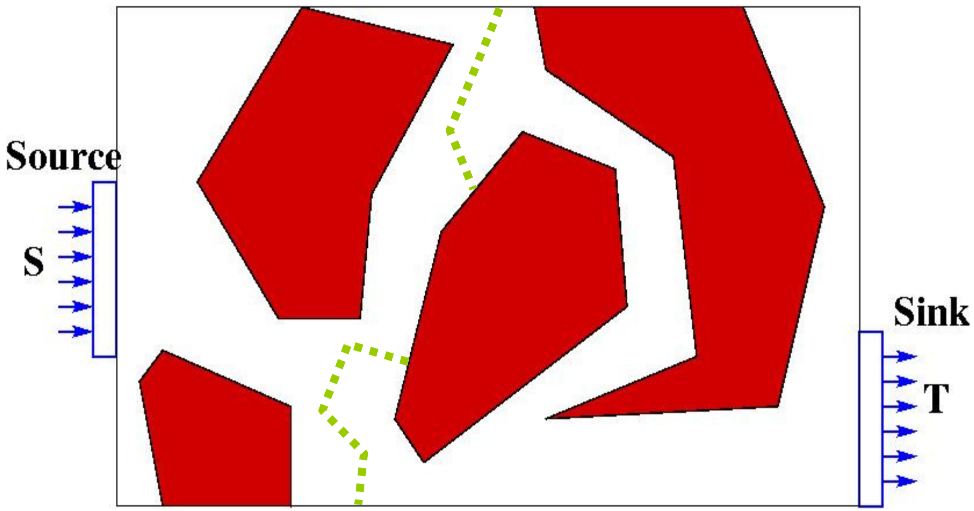
Discrete Network

- Source and sink *nodes*
- Cut
  - partition nodes
  - capacity
    - edges that cross
- Flow
  - integers on arcs



2D Domain

- Source and sink *edges*
- Cut
  - partition domain
  - capacity
    - length of the boundary
- Flow
  - vector field

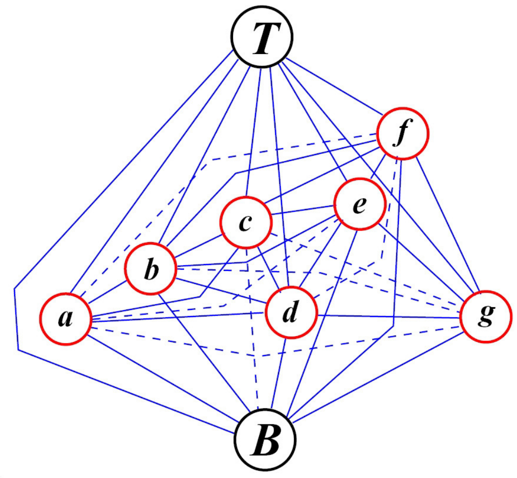


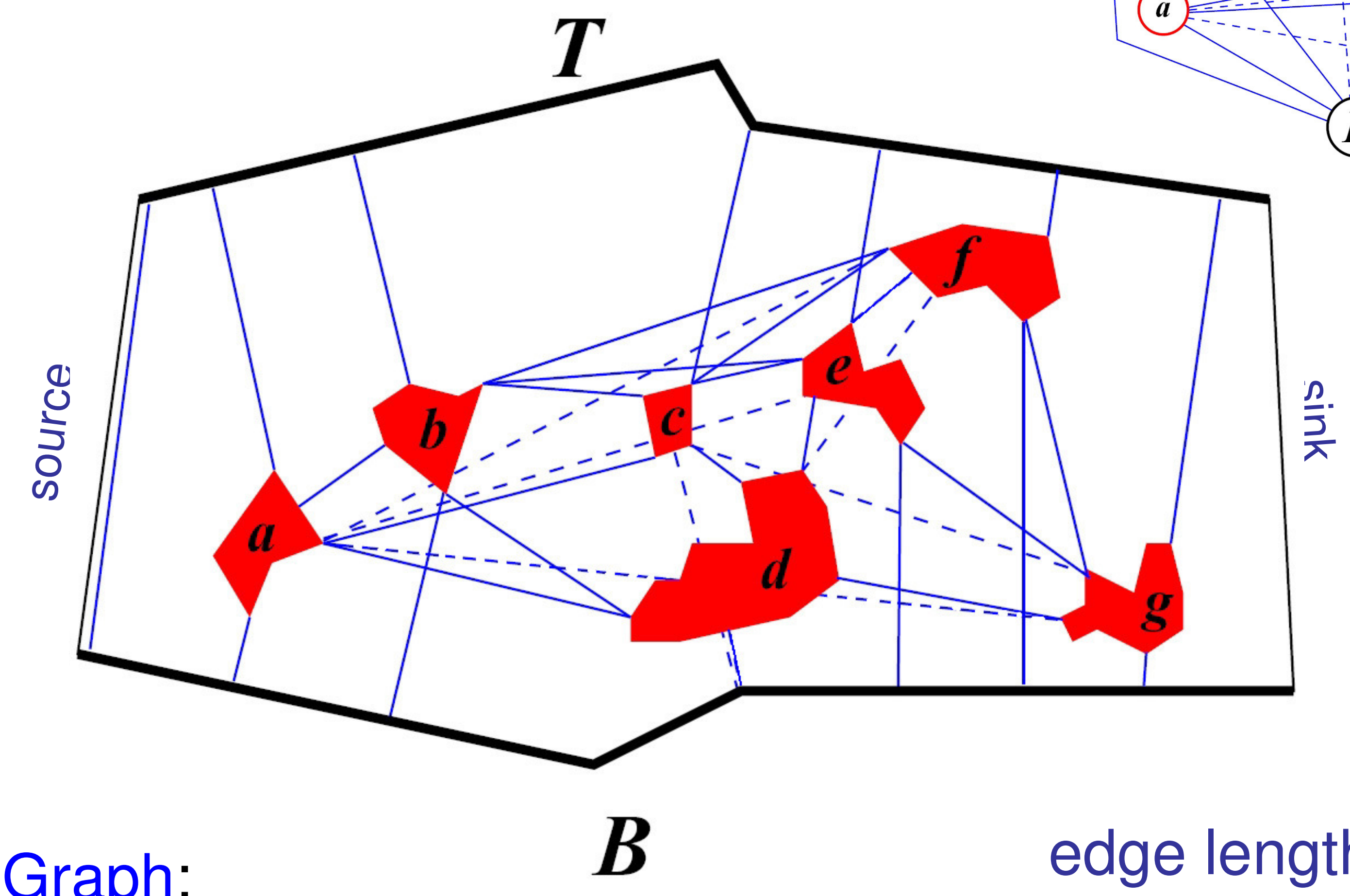
Finding MinCut

Shortest Path in the
“Critical Graph”

Top T, **so** $\leftarrow$ **si**

Bottom B, **so** $\rightarrow$ **si**



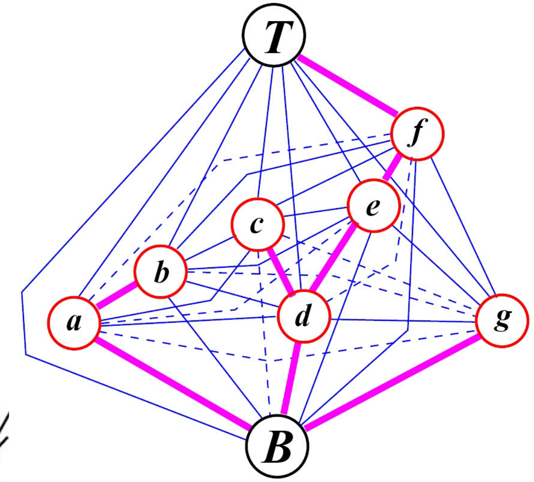


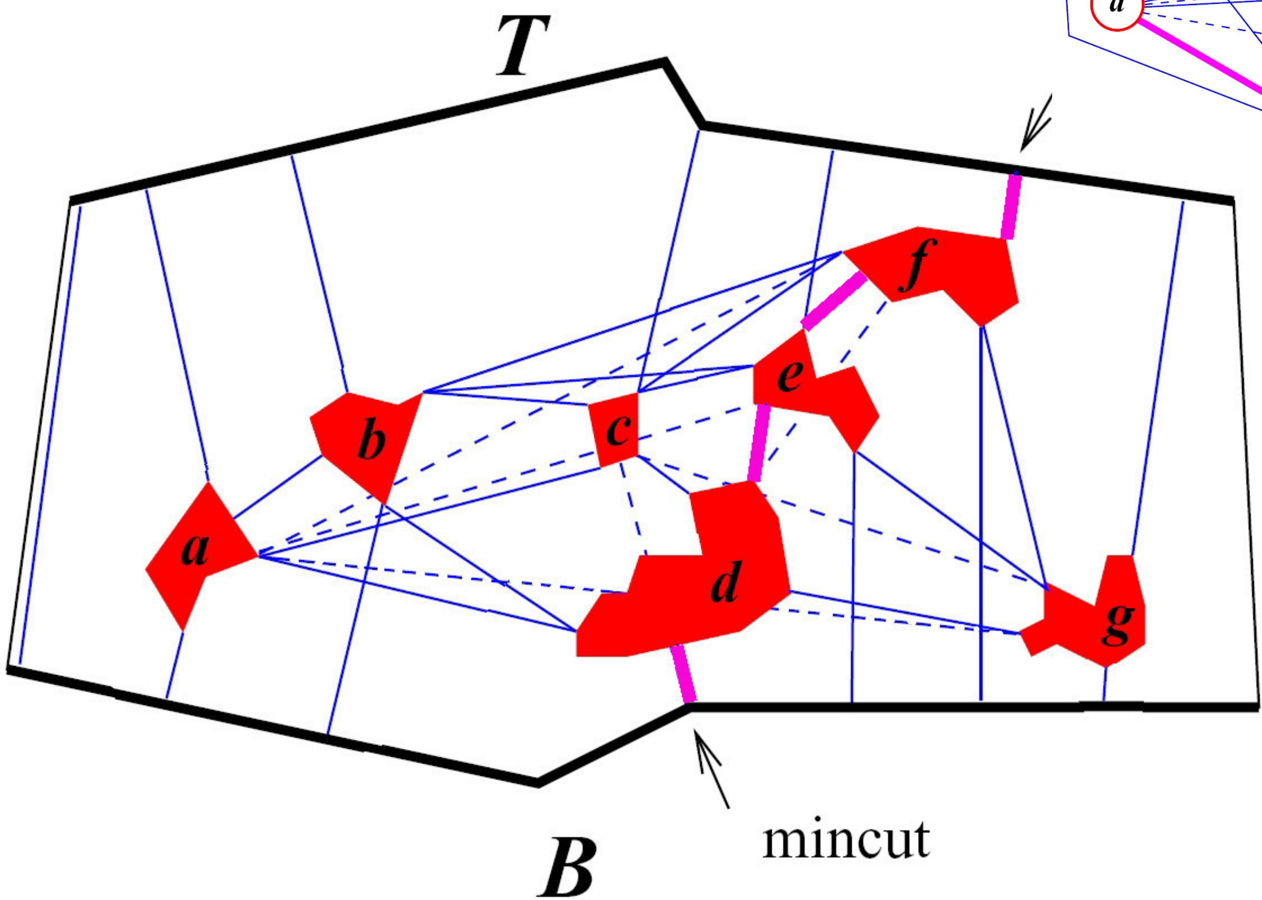
Critical Graph:
nodes for each obstacle, for T, for B

edge length =
Euclidean distance

MinCut =

Shortest ***T-B*** Path in the critical graph





Finding MaxFlow

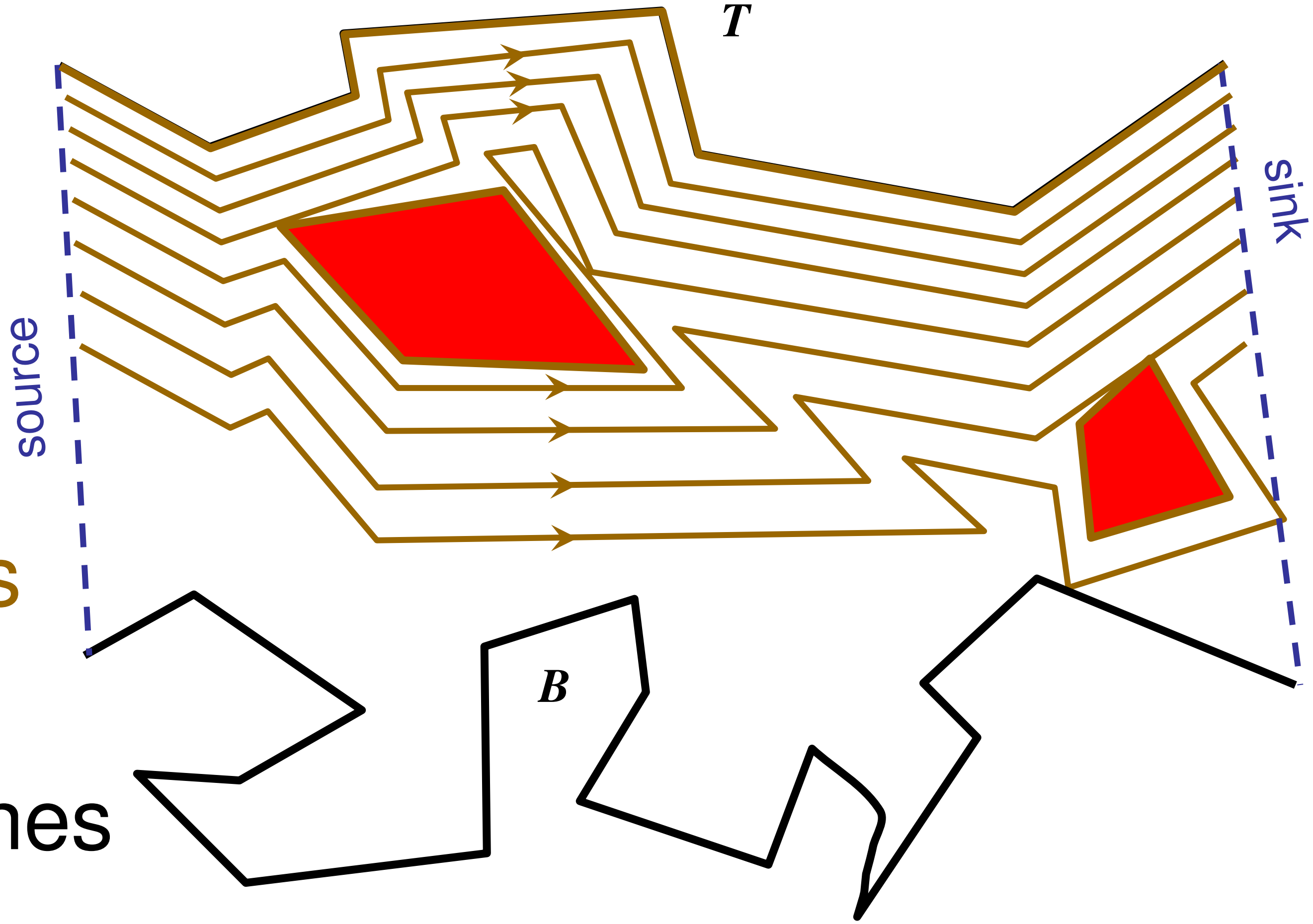
Continuous Uppermost Path Algorithm

- Wave from T
- Wavefront hits a hole
  - continue propagation on hole's other side

Wavefronts

=

Flow streamlines



Continuous MaxFlow/MinCut Theorem

[Mitchell'90]

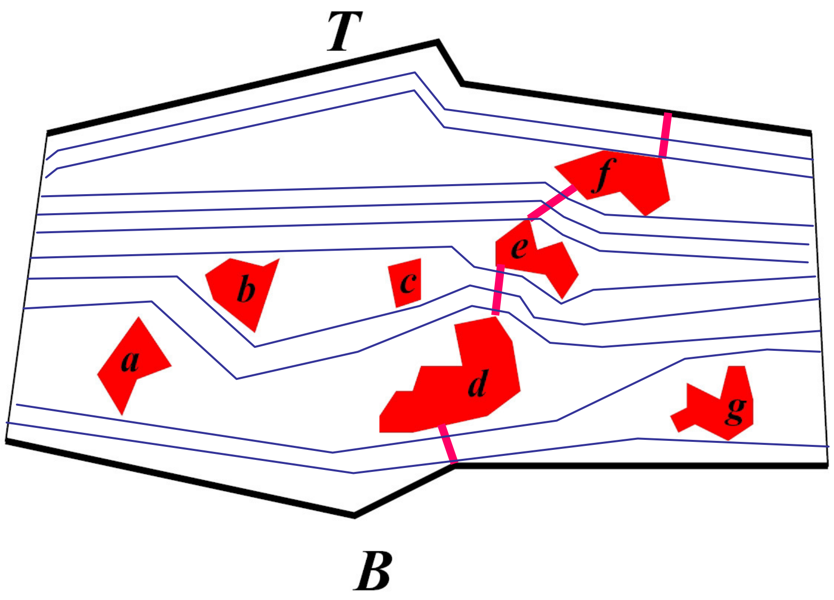
MaxFlow

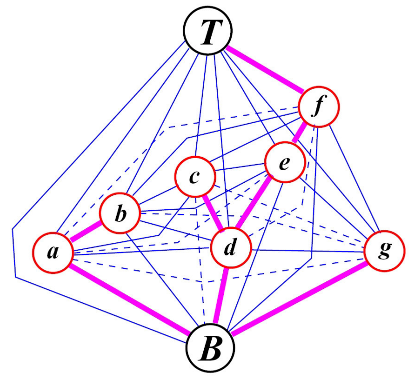
=

MinCut

=

SP T-B path in critical graph

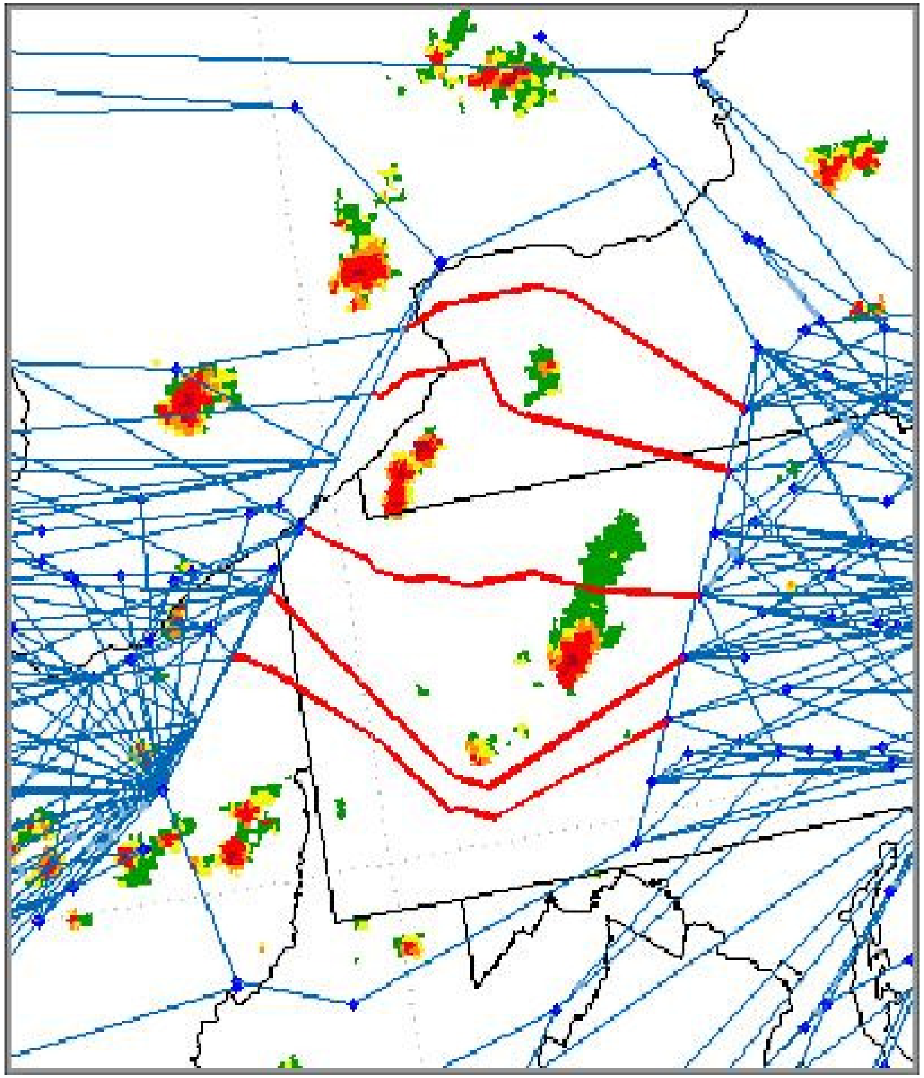




Get Real!

MinCut

Flow Constrained Area (FCA)



Implementation on Real Data

MinCut

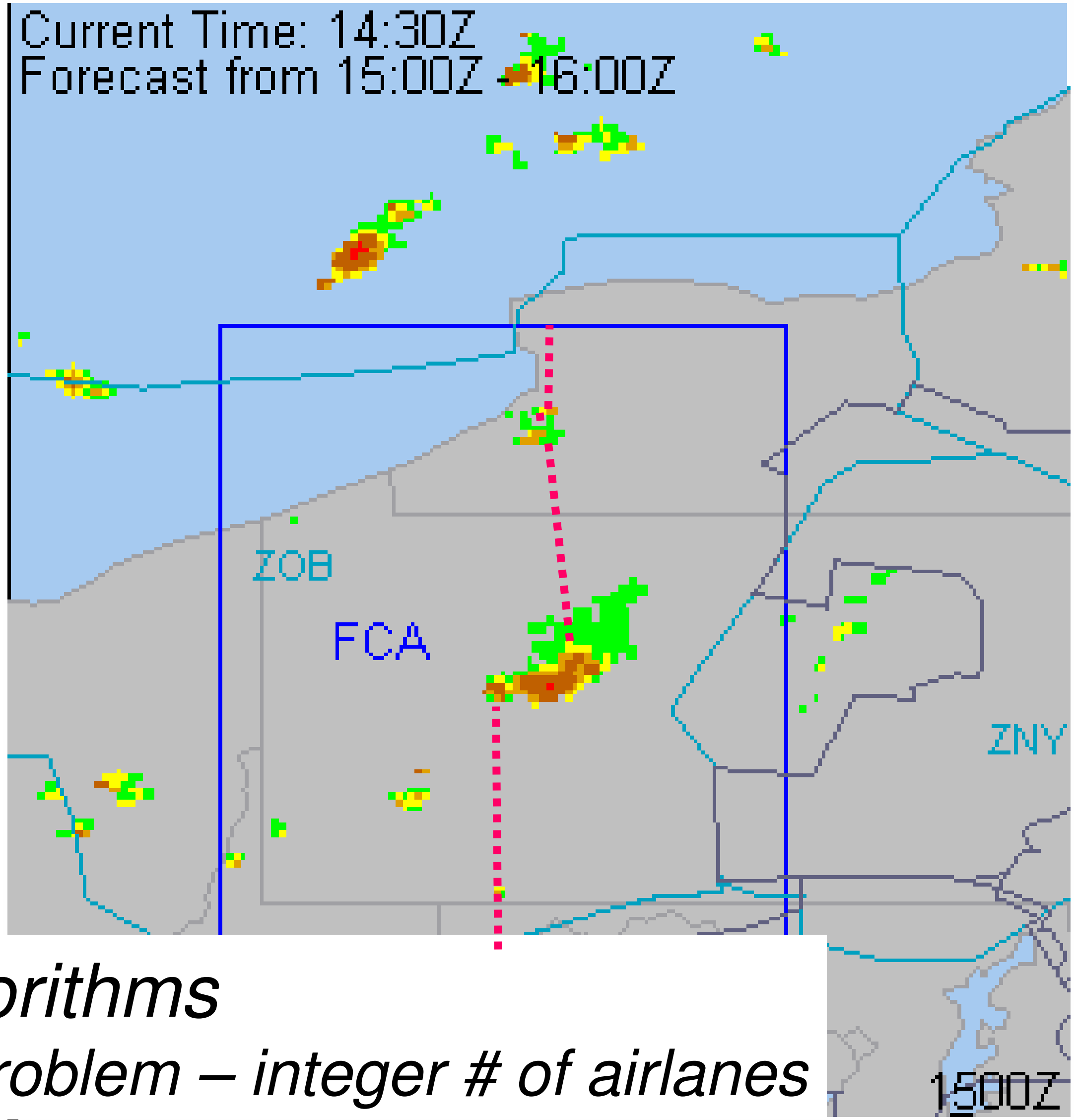
= 21.25 nmi

Airlane width =

5 nmi

Can route

4.25 jets



Modify theory and algorithms

discrete nature of the problem – integer # of airlanes

Continuous Menger's Theorem

Max # of disjoint thick paths

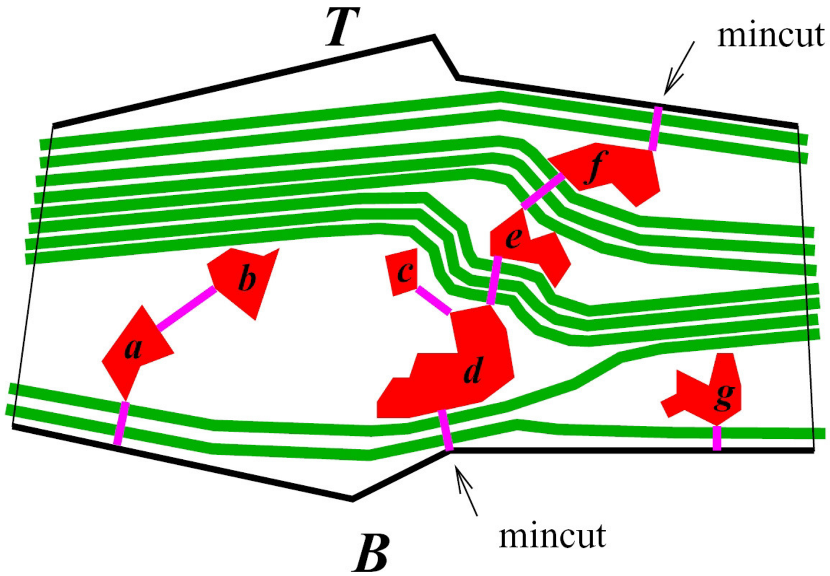
=

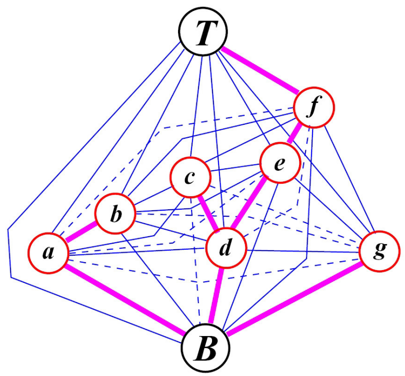
MinCut'

=

SP $\boldsymbol{T}$-$\boldsymbol{B}$ in *thresholded* critical graph

$l_{ij} = \lfloor d_{ij} / \textit{airlane width} \rfloor$

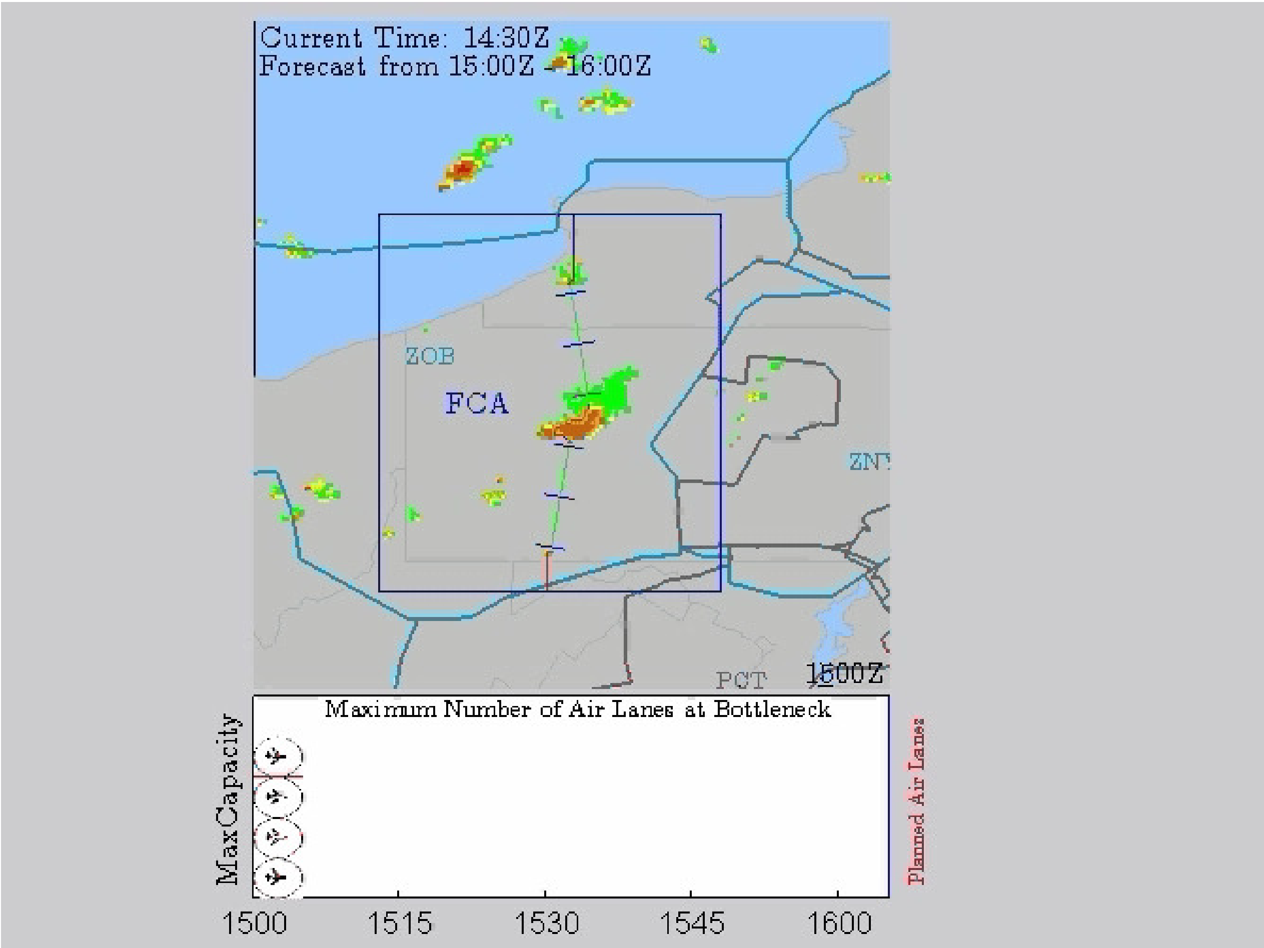




Movie Time!

MinCut

MinCut Over Time

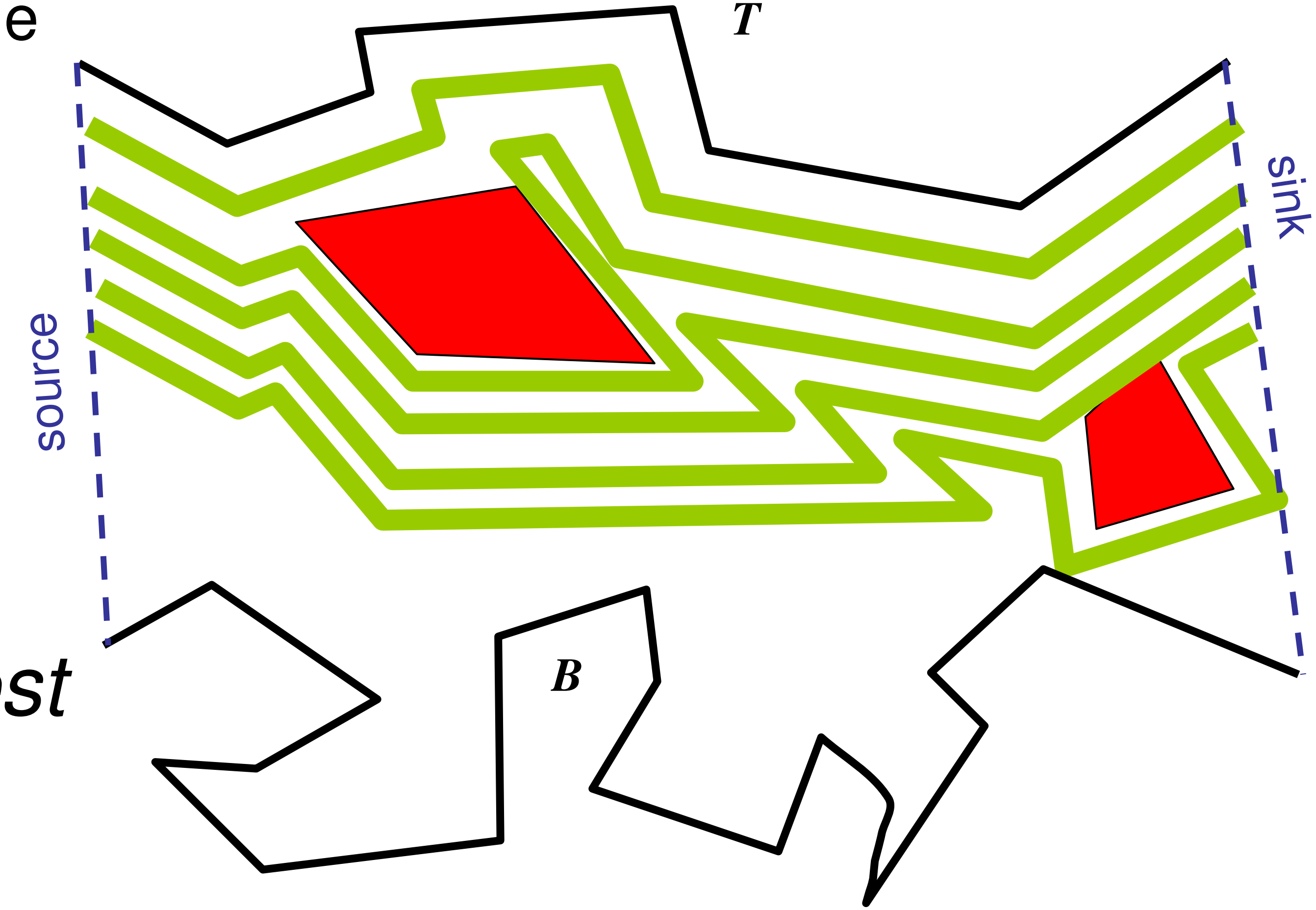


Get Real: Paths…

- “Ugly”, not flyable
- Not short

New task:

Find K *Shortest* Thick Paths

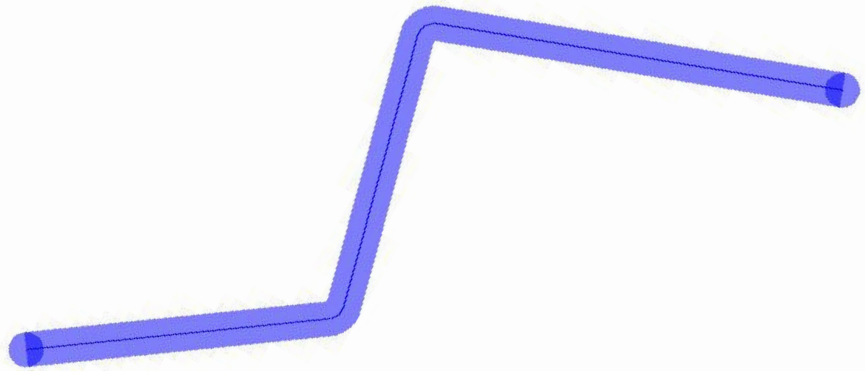


A Thick Path

Thick path =

reference_path $\oplus$ unit_circle

Minkowski sum



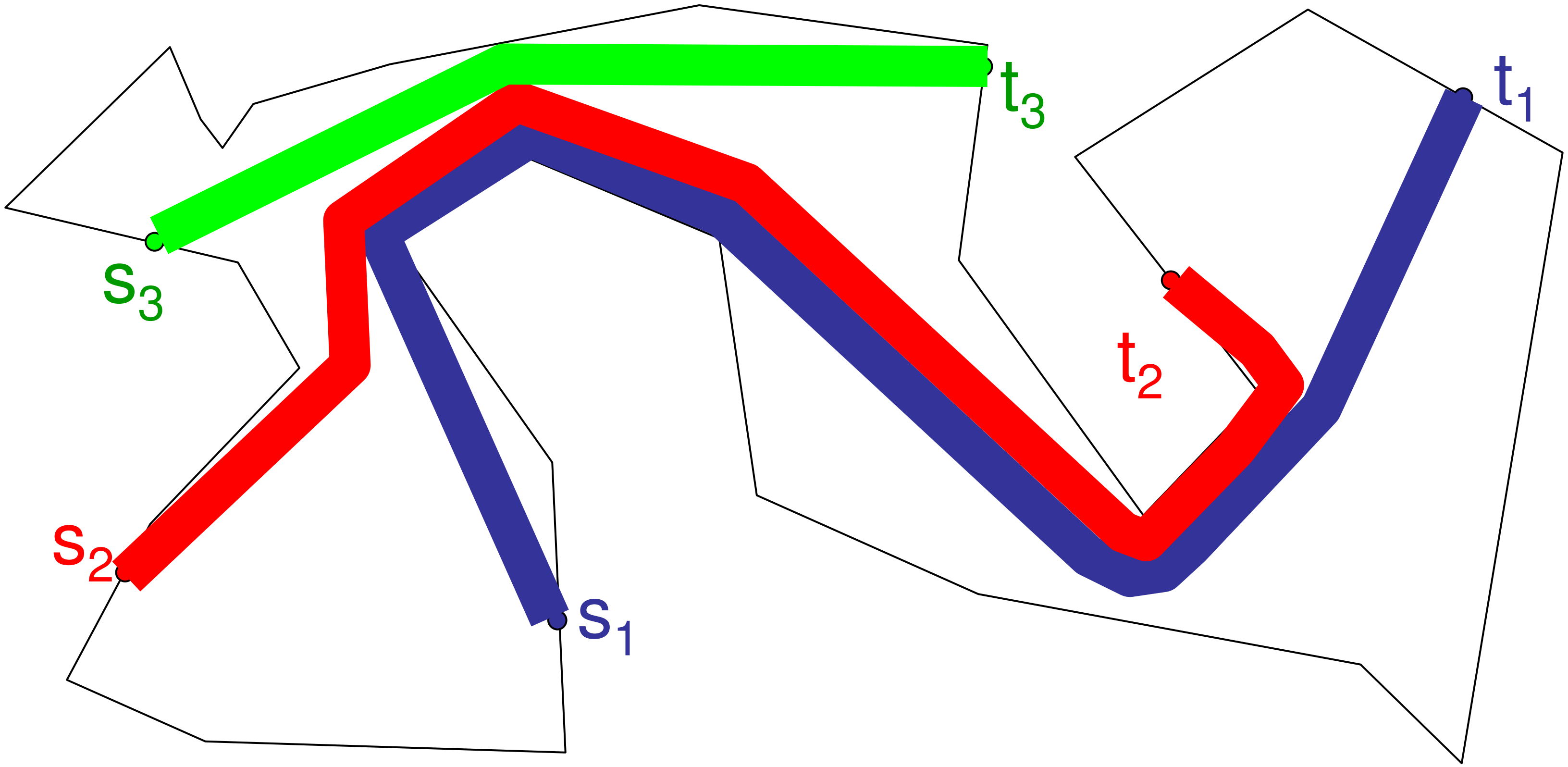
Length =

length(reference_path)

Problem Formulation

- Given: start-destination pairs on bd of P

  { (s_1,t_1), (s_2,t_2), (s_3,t_3) }
- Find: shortest disjoint thickness-2 s_k-t_k paths

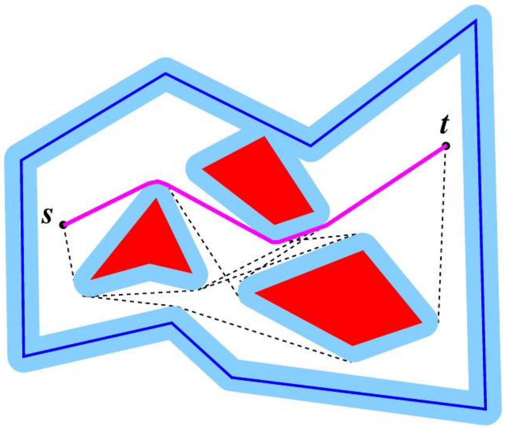


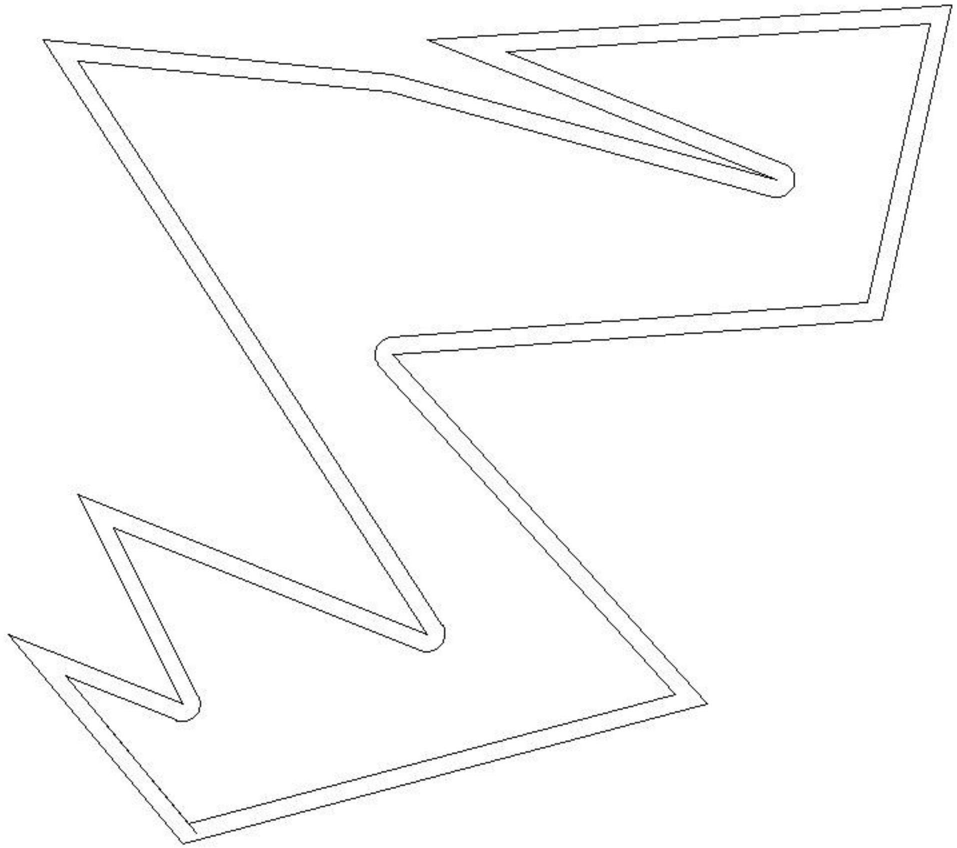
Finding 1 Shortest Thick Path

Inflate by 1

Find shortest path

Inflate the path





2 Paths

Inflate by 1

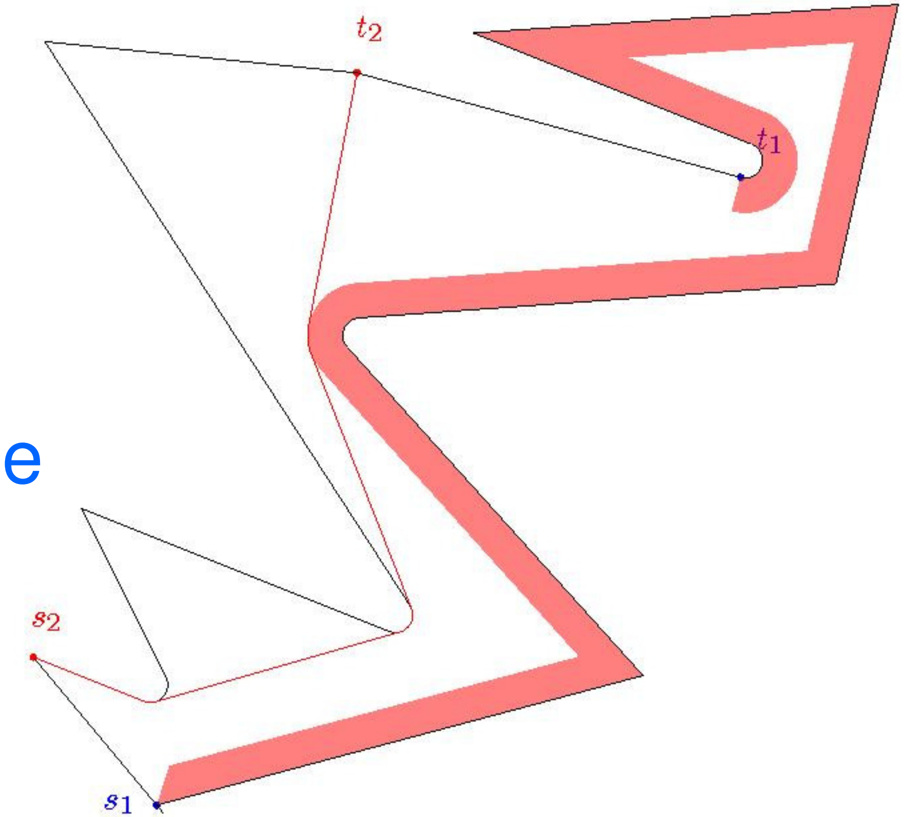
Route s_2-t_2 path

- Inflate $P(s_1,t_1)$ by 2
- Find shortest path

Same on the other side

Inflate the paths

- Non-crossing

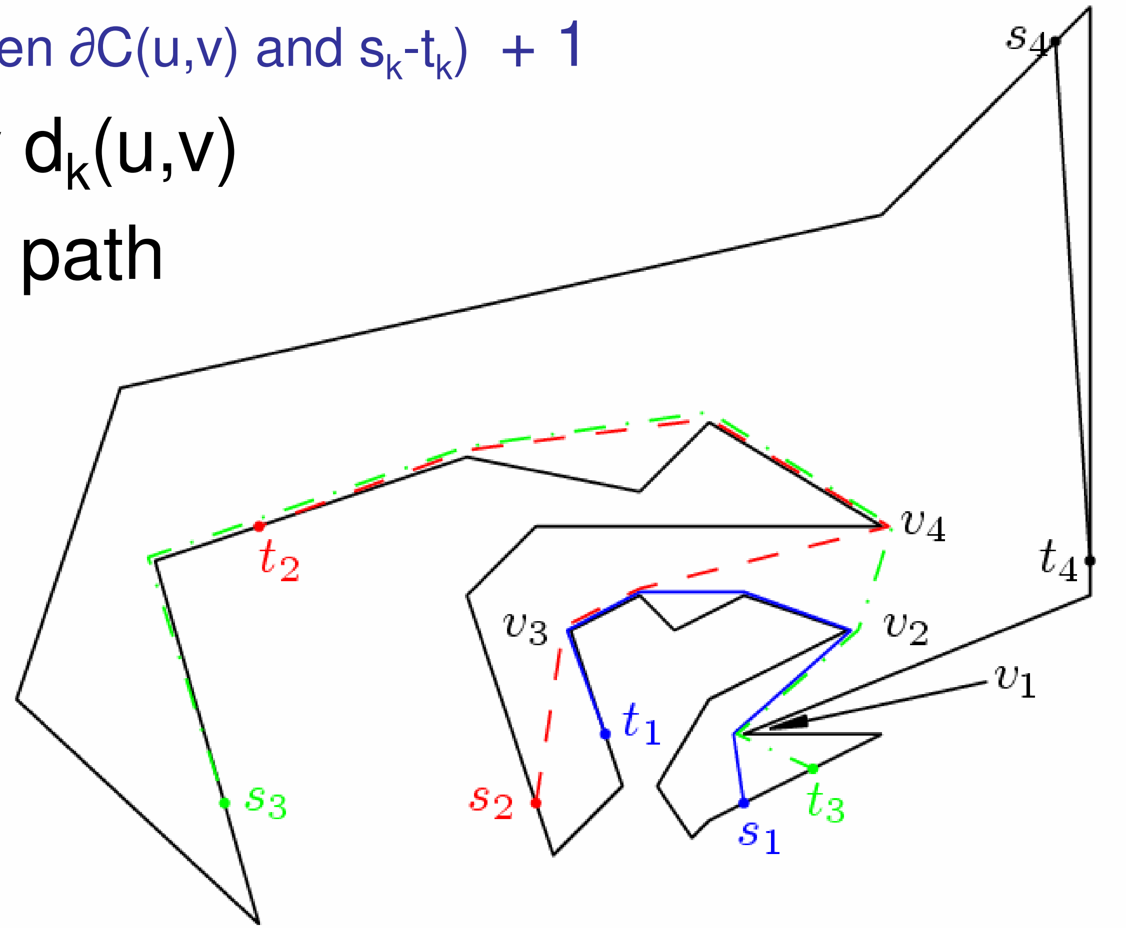


- Each path is ASAP (as short as possible)
  - given the existence of the other
- minsum, minmax

K Paths

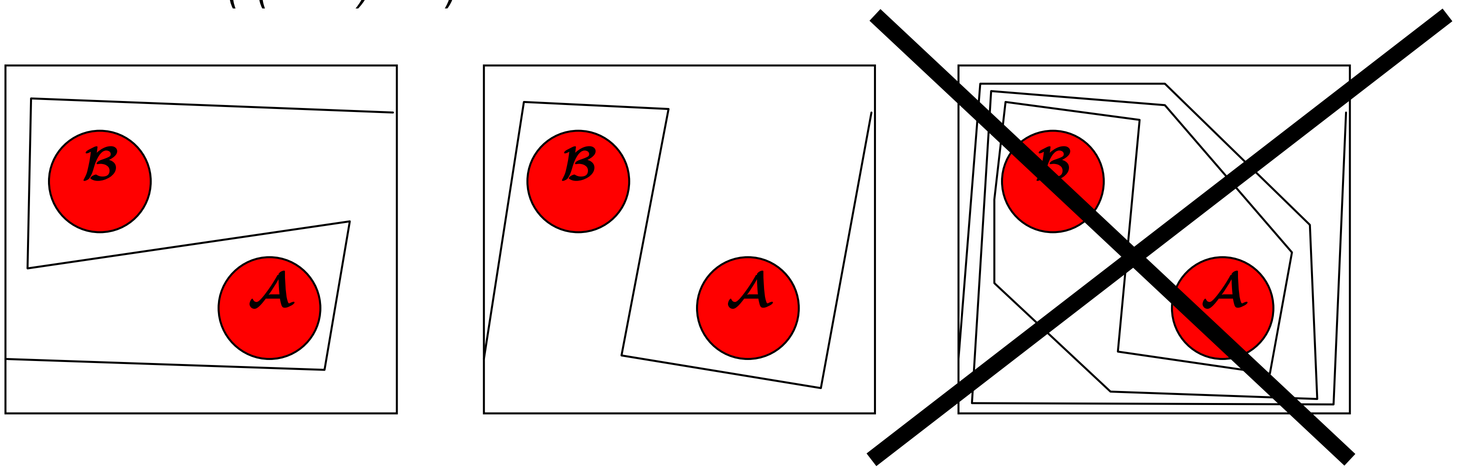
For k = 1…K

- $d_k(u,v)$:= k^{th} depth of $\partial C(u,v)$ ≡
- ≡ 2 • (# of paths between $\partial C(u,v)$ and s_k-t_k) + 1
- Inflate $\partial P(u,v)$ by $d_k(u,v)$
- Find the shortest path
- Inflate the path



Polygons with Holes

- # of holes h is large
  - NP-hard
- $h = O(1)$
  - Scroll through relevant homotopy types
    - $O((K+1)^h h!)$ of them



$O((K+1)^h h! \text{ poly}(n,K))$

fixed-parameter tractable

[Rod and Fellows '99]

“Gluing” Shortest Thick Paths

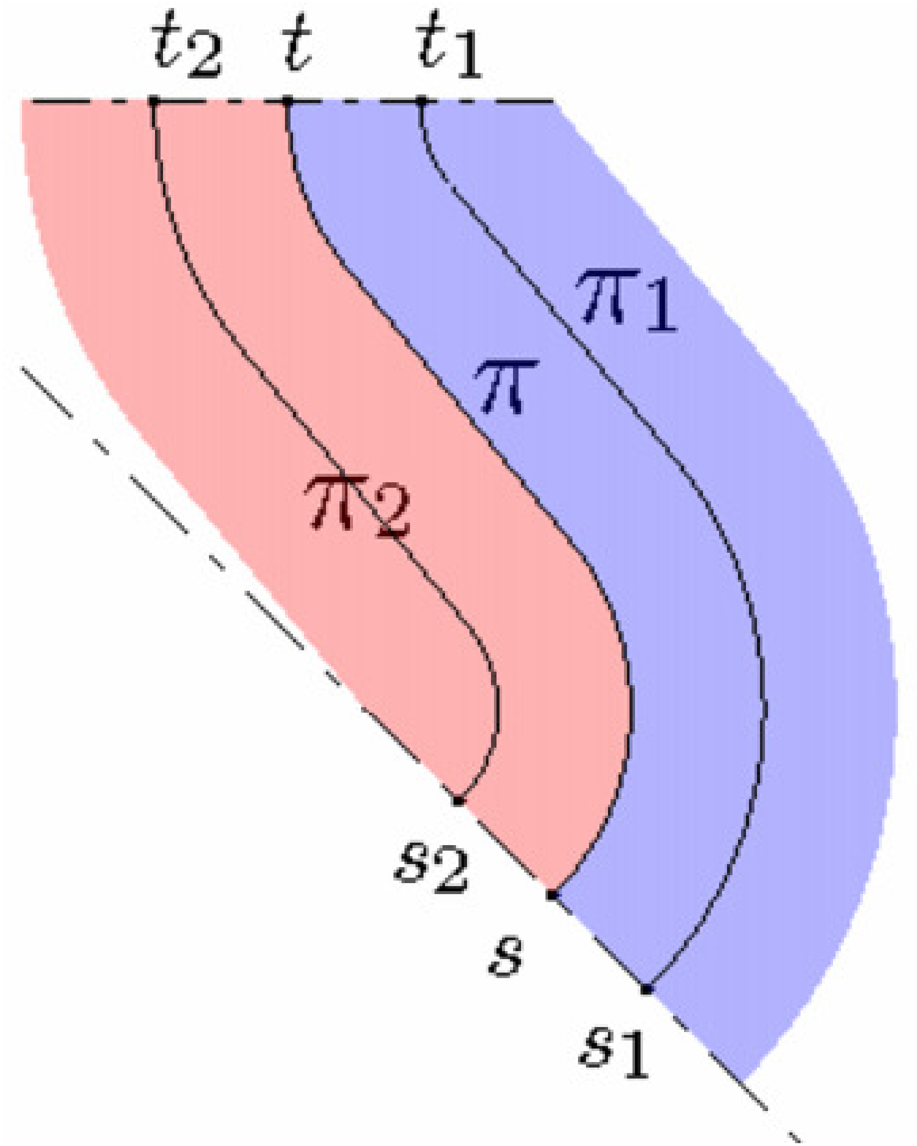
SP s-t of width 4

=

SP s_1-t_1 of width 2

+

SP s_2-t_2 of width 2



From Paths to Flows

Min-Cost Flow

Given: Polygonal domain P

sources **S** and sinks **T**

Flow

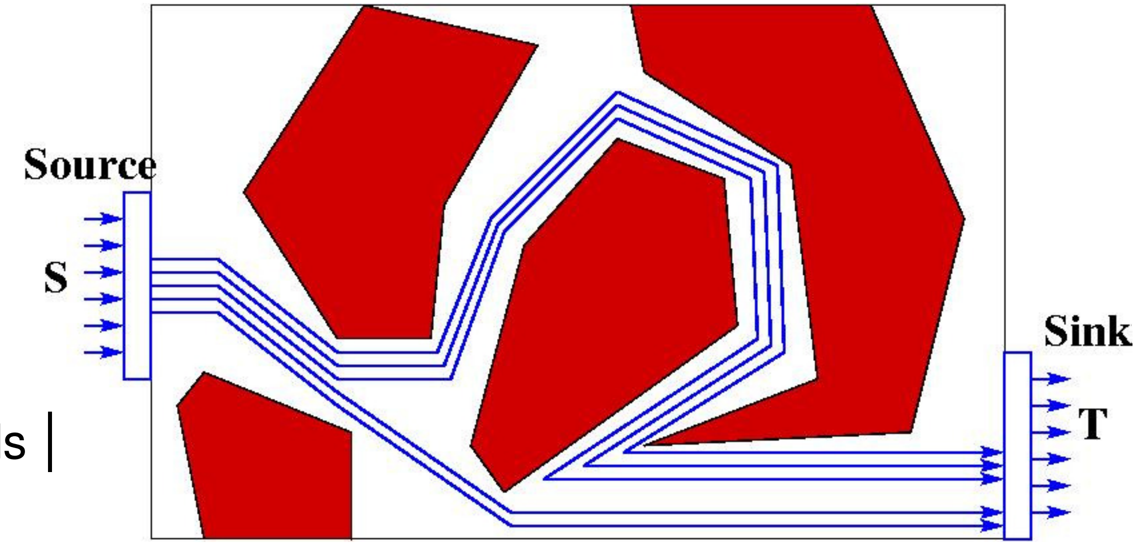
vector field σ

div $\sigma = 0$ inside P

$\sigma \cdot n = 0$ on $\partial P \setminus \{\mathbf{S},\mathbf{T}\}$

$|\sigma| \le 1$ – capacity

$V = \left| \int_{\mathbf{S}} \sigma \cdot n \, ds \right| = \left| \int_{\mathbf{T}} \sigma \cdot n \, ds \right|$



Min-Cost Flow

Given V

Find σ that minimizes *cost*

Cost

$|l_s|$ – length of streamline through s in **S**

$$\text{cost} = \int_{\mathbf{S}} |l_s| \, ds$$

Equivalent Problems

l_s – streamline of σ through s in S

$l_s: \mathbf{S} \rightarrow \mathbf{T}$

σ_{ab} – restriction of σ

Split a into s_k, k=1…K

Split b into t_k, k=1…K

SP(s_k,t_k), width w/K

Path $\rightarrow l_s$ as K $\rightarrow \infty$

Min-cost $\sigma_{ab} = \int_a |l_s| ds$

$= \Sigma_k SP(s_k,t_k)$

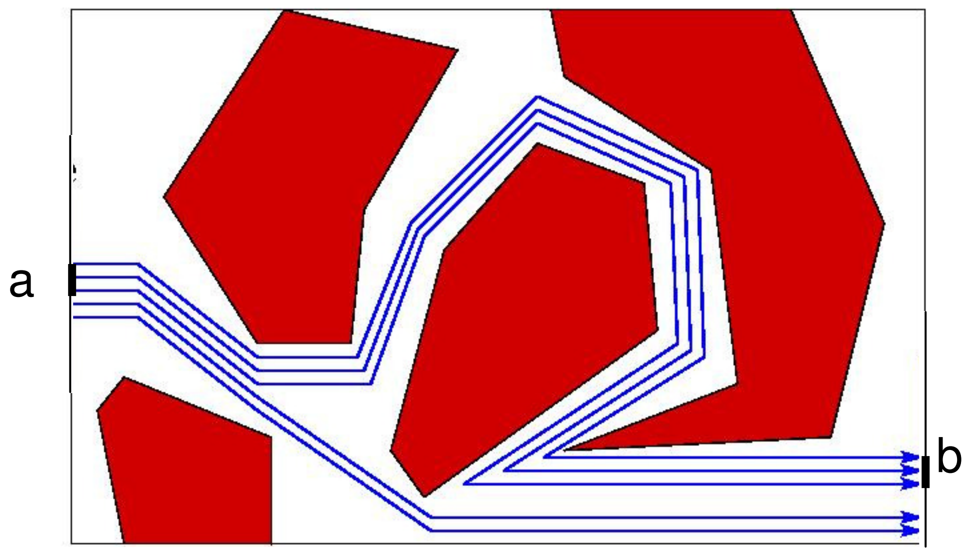
Contiguous subsets of **S**

mapped into

Contiguous subsets of **T**

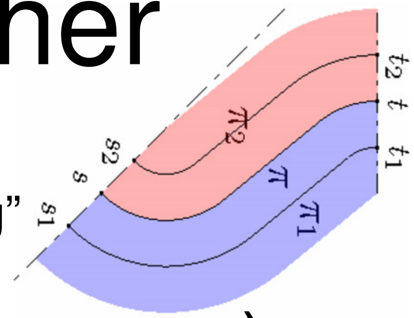
$l_s(a) = b$,

$|a| = |b| = w$



Gluing Things Together

Min-cost $\sigma_{ab} = \Sigma_k SP(s_k,t_k) =$ by "Gluing"



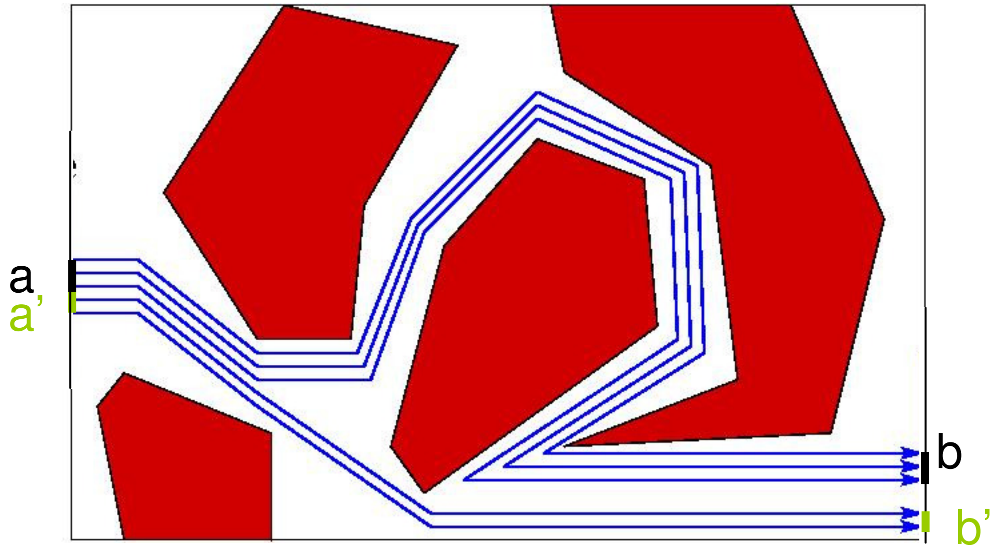
= 1 Shortest Thick Path (from midpoint of a to midpoint of b)

$\sigma = \sigma_{ab} + \sigma_{a'b'} + \sigma \ldots$

Intervals of continuity of l_s: **S**→**T** and its inverse

⇓

Min-cost Flow = Thick Non-Crossing Paths



The Continuous Flow Decomposition Theorem

The support of

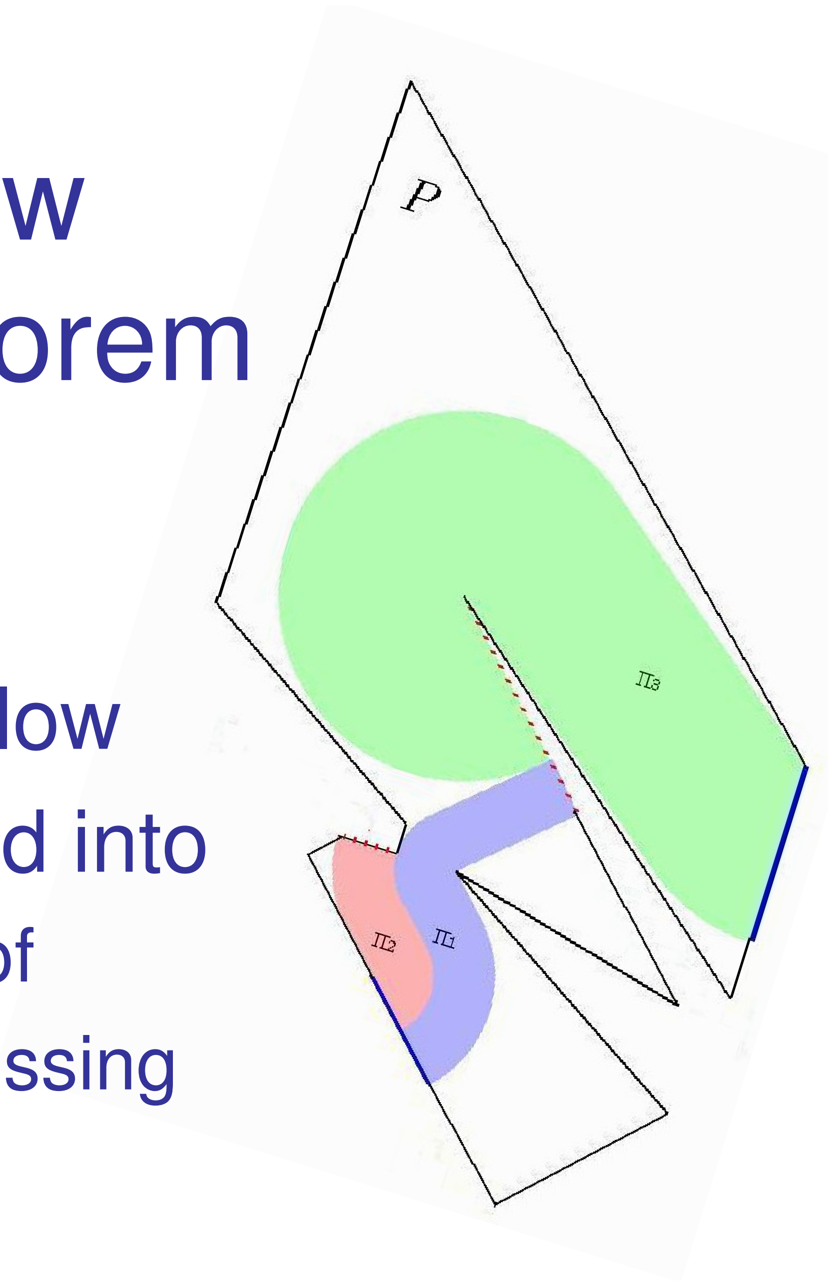
a minimum-cost flow

may be decomposed into

a linear number of

shortest thick non-crossing

paths



Get Real!

Thick Paths

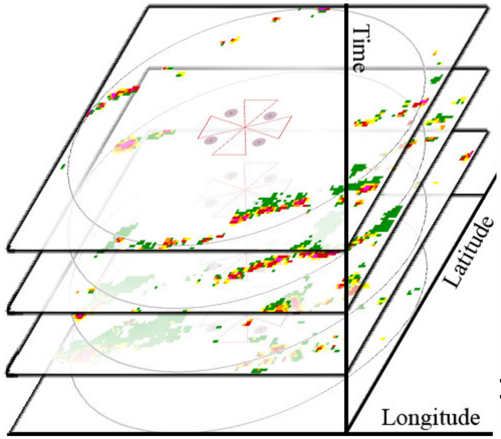
Flow-Based Route Planner

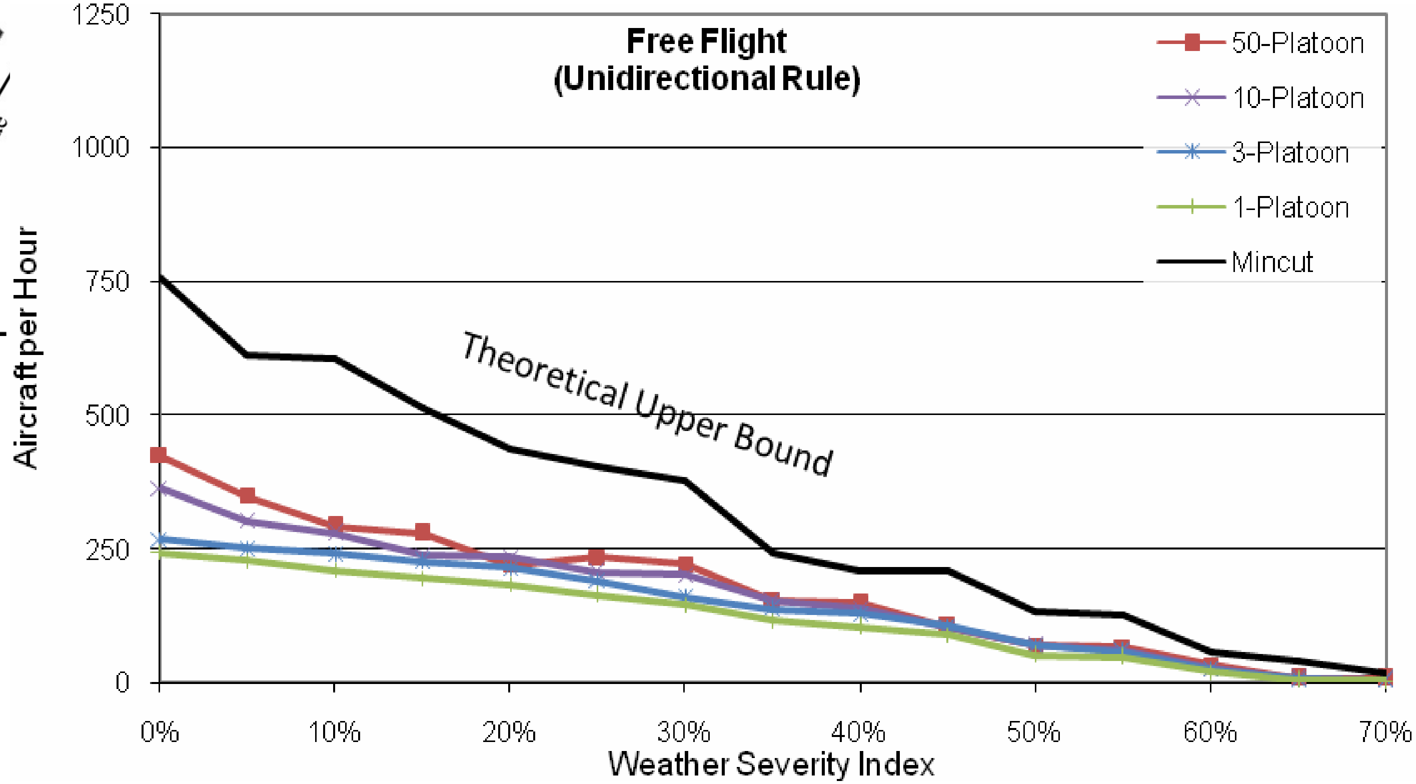
[Prete '07,

Krozel, Mitchell, Penny, P, and Prete '04-07]

Flow-Based Route Planner

- Heuristic
  - Uniform grid
- While possible
  - find a path
  - call it an obstacle

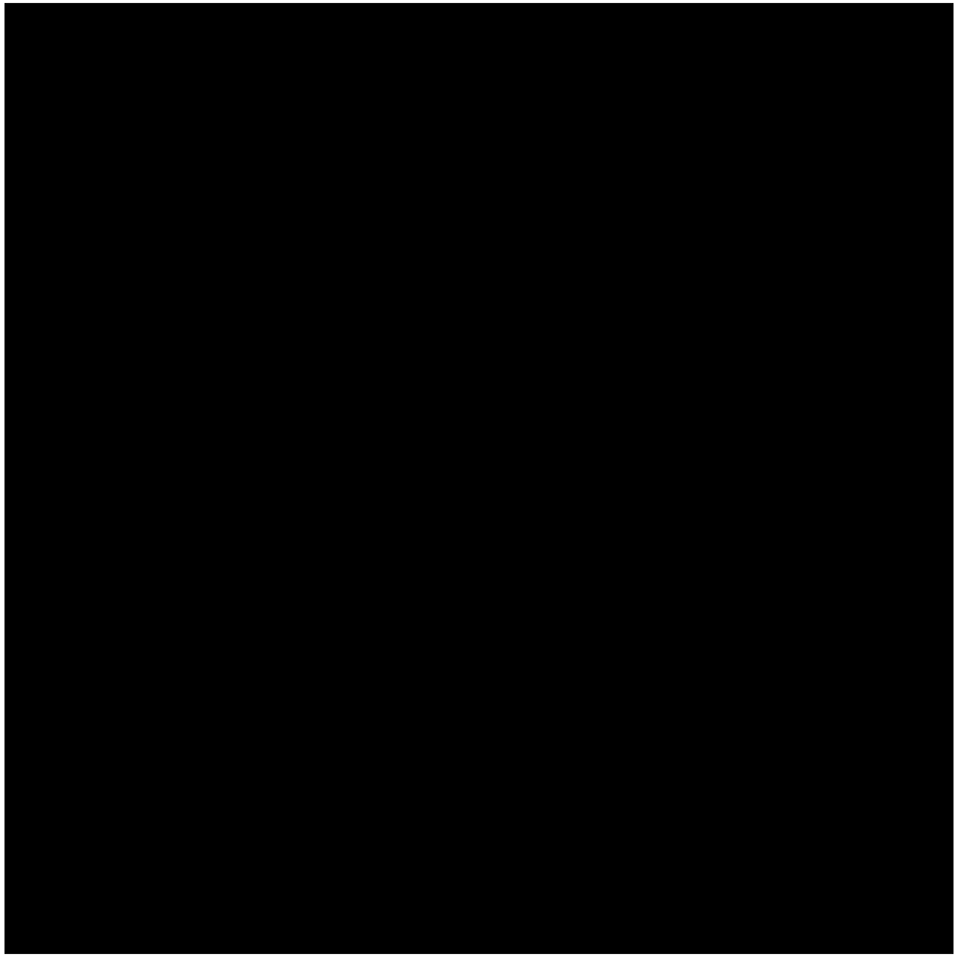




Movie Time!

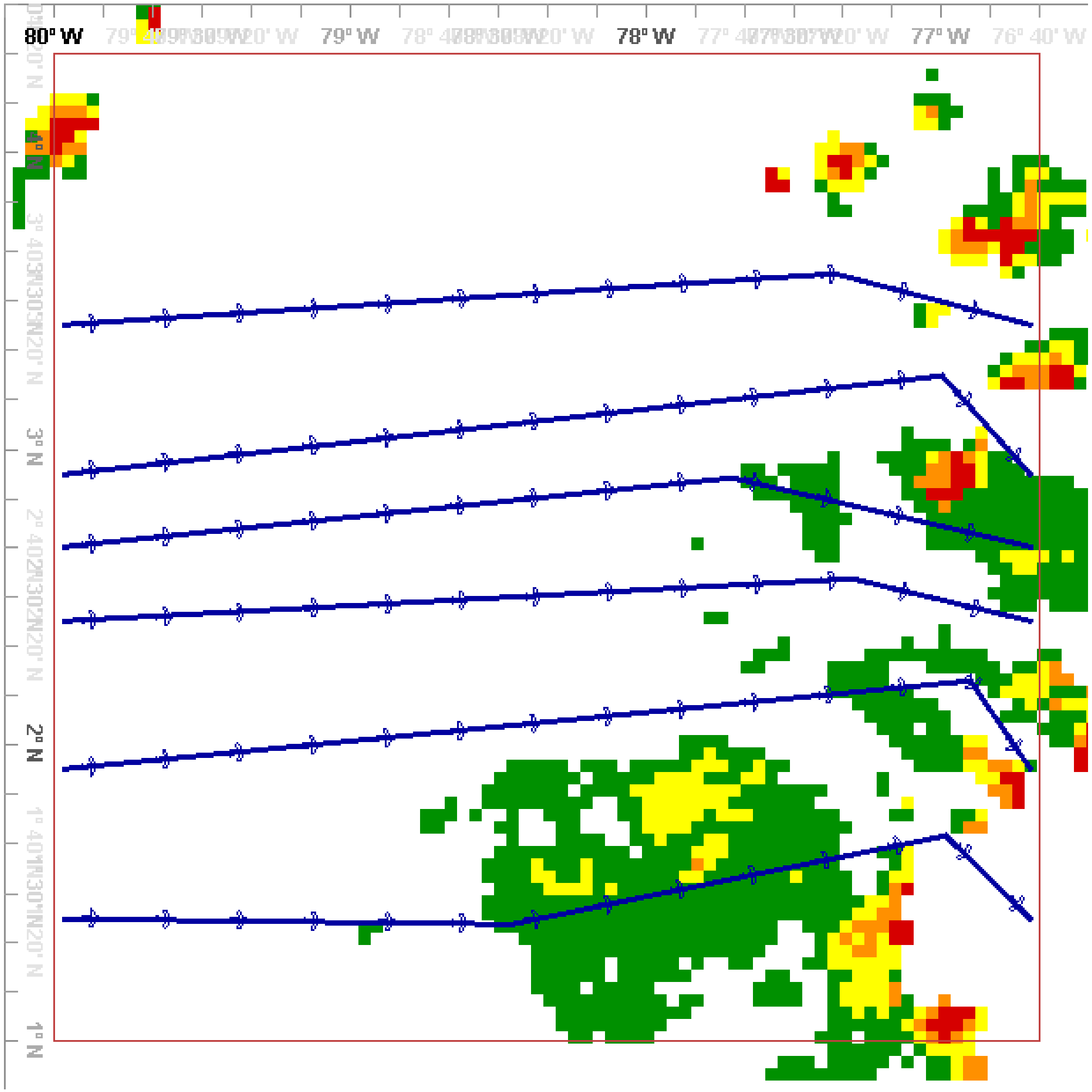
Thick Paths

Flow-Based Route Planner



(At Least) Two Shortcomings

- Obstacles are static
- The routes are non-crossing



Dynamic Version

- Given
  - *moving* obstacles
- Find
  - trajectories for a maximum number of aircraft

Movie Time!

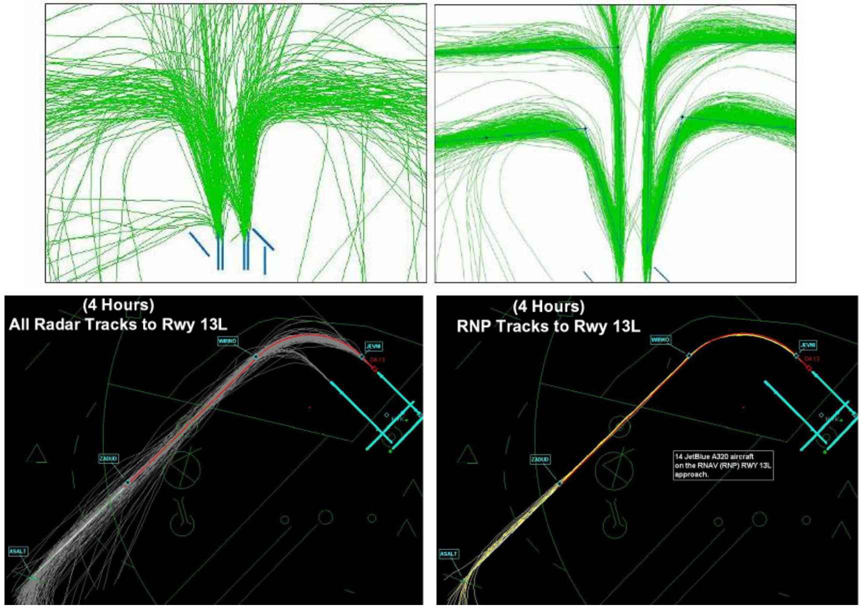
Thick Paths in Dynamic Environment

Dynamic Planning

If there exist **K** paths for
unit discs
moving with speed **≤ 1**

we find, for any $\Delta t < ½$

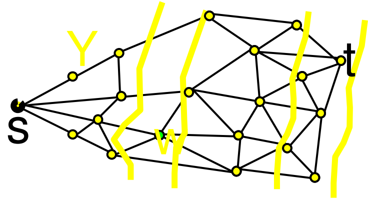
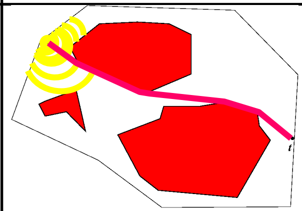
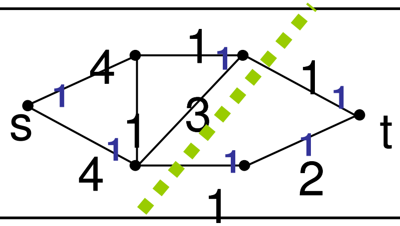
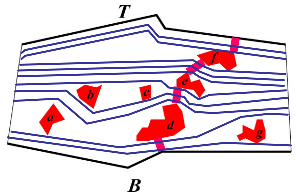
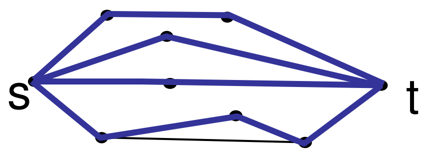
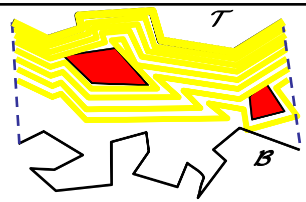
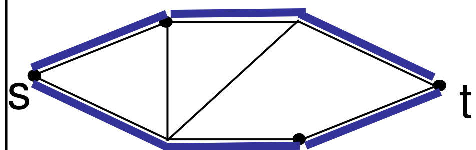
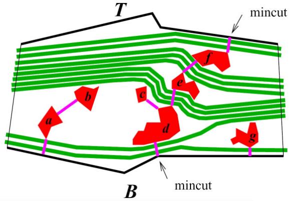
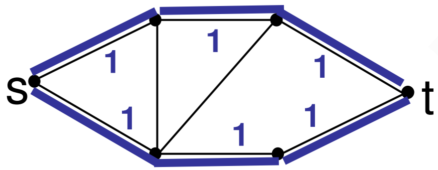
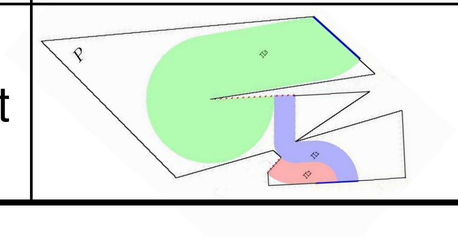
≥ K paths for
discs of **radius** $\mathbf{(1/4 - 3/8 \cdot \Delta t)^2}$
moving with speed **≤ 7 / Δt**



Improve navigation and speed – get opt!

Path thickness = navigation performance

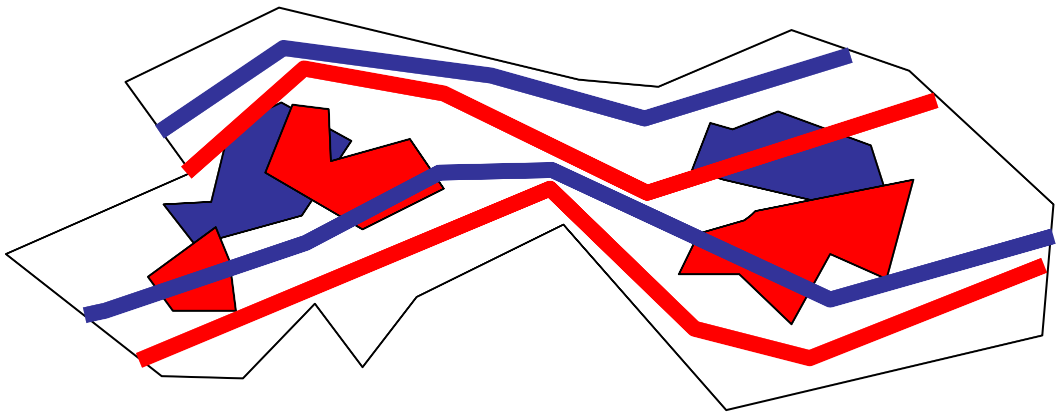
Summary

	Networks	2D
Dijkstra's Alg [Mitchell'86]		
MaxFlow/MinCut [Strang'83]		
Uppermost Path Alg [Mitchell'90]		
Menger's Theorem [Arkin, Mitchell, and P '08]		
Flow Decomposition [Mitchell and P '07]		

Open: Theory

- SP on surface of arbitrary polyhedron
- 1 thick path, $o(n^2)$ alg
- 1 thick wire, poly-time alg

- Planning in face of uncertainty
- Geometric multicommodity flows



Open: Practice

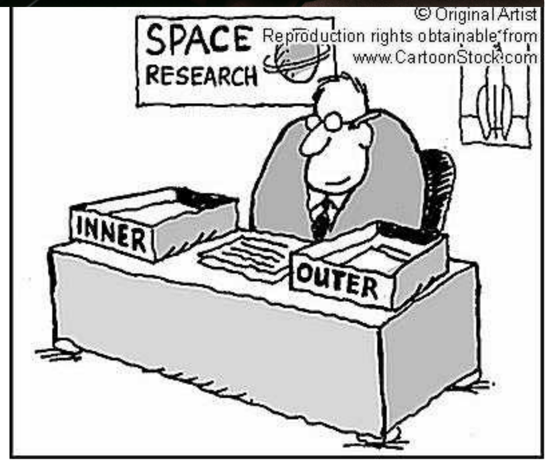
- Implement existing algs
- Design new

Produce movies!

Pilots

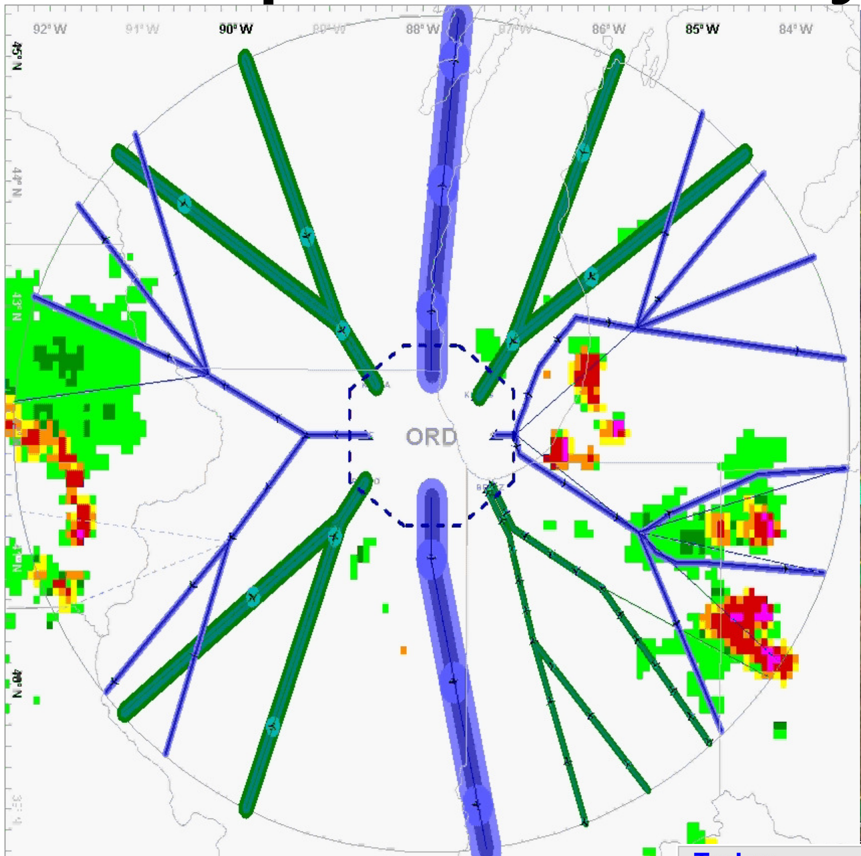
ATC

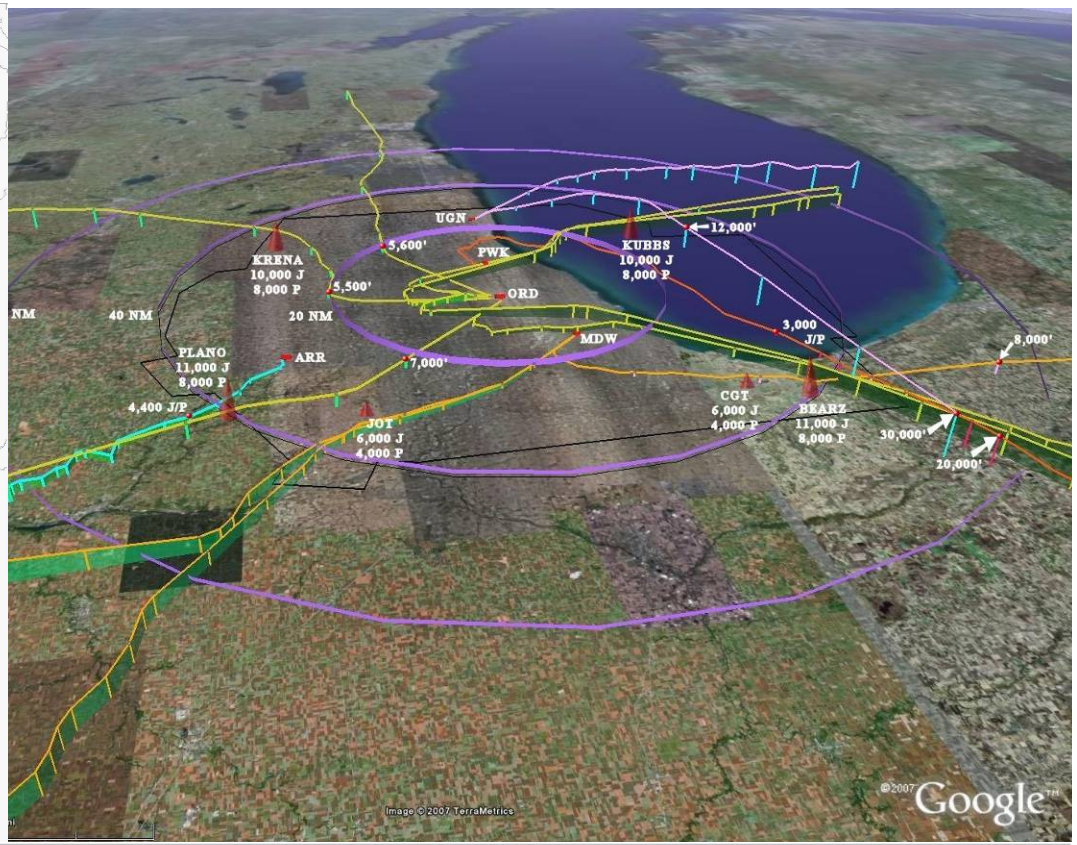
ATM

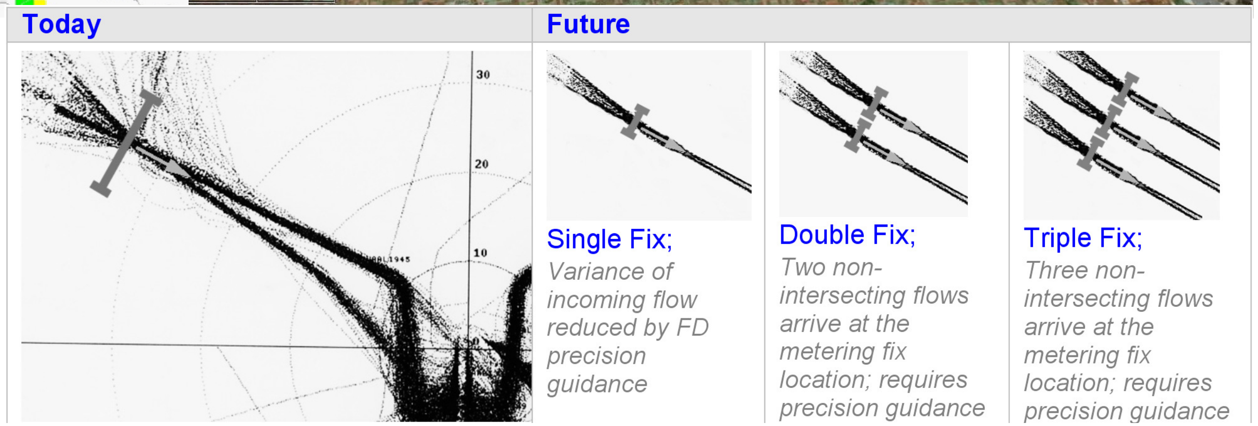


Researchers

Open: Theory and Practice



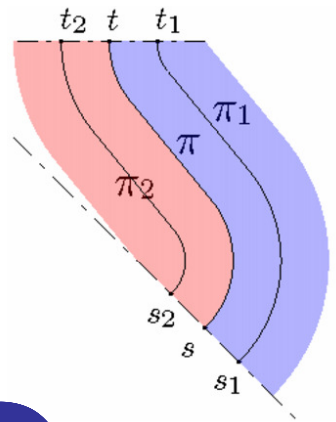




Arrival/departure trees

Constrained paths

(turns, wiggle rooms)



Thank you!