Further topics (with possible papers to suggest):

- routing
- Travelling Salesman Problem (today)
- Frechet distance and polygon sweeping
- reconfiguration
- online shortest paths
- thick paths, airspace problems
- betweenness centrality
- etc.

Deadline for selecting a paper or suggesting a topic from above: Wednesday October 15.

Schedule:

Mon Oct 6
Wed Oct 8 Hamide

Mon Oct 13 - Thanksgiving
Wed Oct 15 Mustaq Ahmed

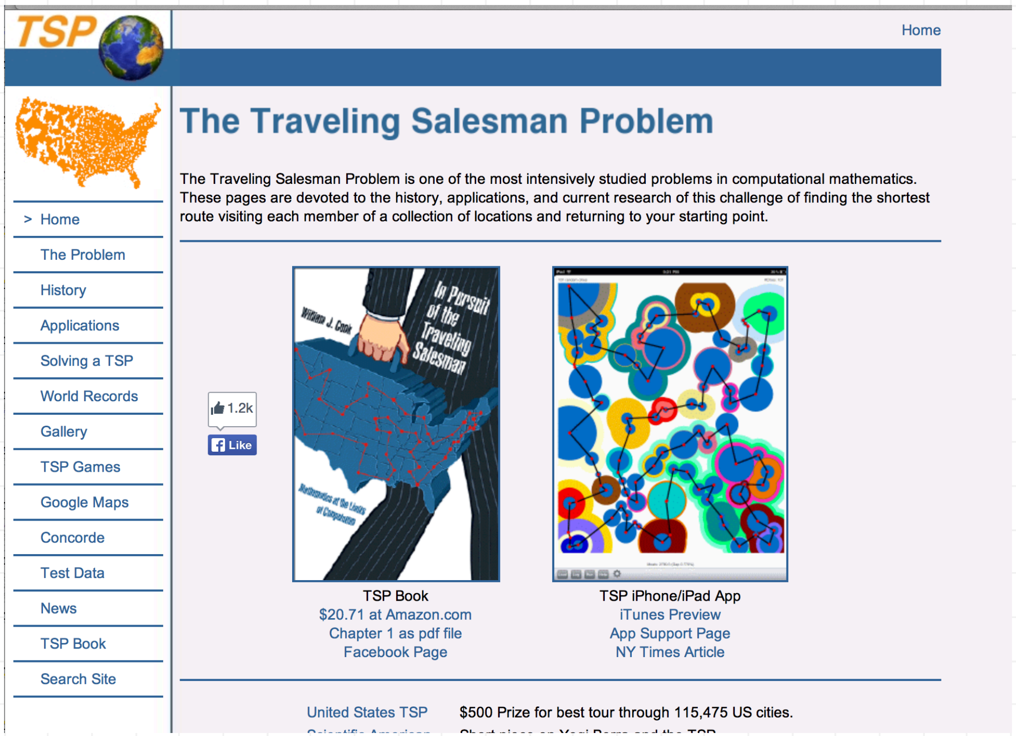
Mon Oct 20 Philippe Nathan (or Wed.)
Wed Oct 22 Khaled

Mon Oct 27
Wed Oct 29

Shortest paths through multiple targets — Travelling Salesman and etc.

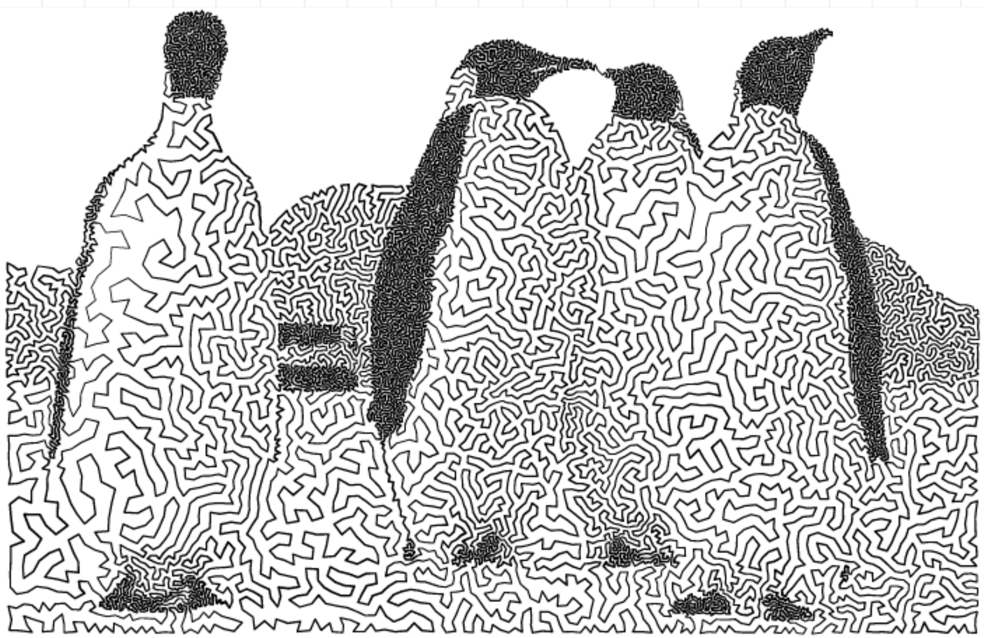
Travelling Salesman Problem. Given a graph G=(V,E) with weights on edges $w: E \to R^{\geq 0}$, find a TSP tour — a cycle C that visits every vertex exactly once and has minimum weight $\sum_{e \in C} w(e)$

Bill Cook, U. Waterloo http://www.math.uwaterloo.ca/tsp/



Travelling Salesman Problem. Given a graph G=(V,E) with weights on edges $w: E \to R^{\geq 0}$, find a TSP tour — a cycle C that visits every vertex exactly once and has minimum weight $\sum_{e \in C} w(e)$

Craig Kaplan



Travelling Salesman Problem.

TSP is NP-complete

Approximation algorithms:

general ⊇ metric ⊇ Euclidean

general: no constant-factor approx (unless P=NP)

metric: 1.5 approx → recent progress

Euclidean: vertices are pts. in plane, distance = Euclidean

PTAS - polynomial time approx scheme

'98 - Mitchell
- Arora

↑ could present

Approximating graphic TSP by matchings
T Momke, O Svensson - Foundations of Computer Science (…, 2011 - ieeexplore.ieee.org
Abstract—We present a framework for approximating the metric TSP based on a novel use of matchings. Traditionally, matchings have been used to add edges in order to make a given graph Eulerian, whereas our approach also allows for the removal of certain edges ...
Cited by 54 Related articles All 21 versions Cite Save

From: http://scholar.google.ca/scholar?hl=en&q=M%C3%B6mke+and+Svensson&btnG=&as_sdt=1%2C5&as_sdtp=

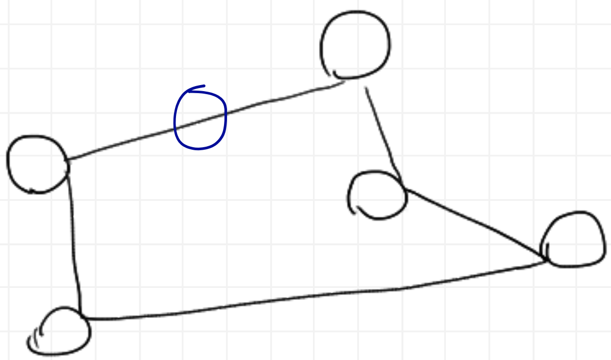
↖ + related papers

Travelling Salesman Problem with neighbourhoods.

[Approximation algorithms for TSP with neighborhoods in the plane](http://scholar.google.ca/scholar?cluster=2201776311898066735&hl=en&as_sdt=0,5)
A Dumitrescu, JSB Mitchell - Journal of Algorithms, 2003 - Elsevier
In the Euclidean TSP with neighborhoods (TSPN), we are given a collection of n regions (neighborhoods) and we seek a shortest tour that visits each region. As a generalization of the classical Euclidean TSP, TSPN is also NP-hard. In this paper, we present new ...

From: http://scholar.google.ca/scholar?cluster=2201776311898066735&hl=en&as_sdt=0,5

+ later papers



Touring a sequence of polygons

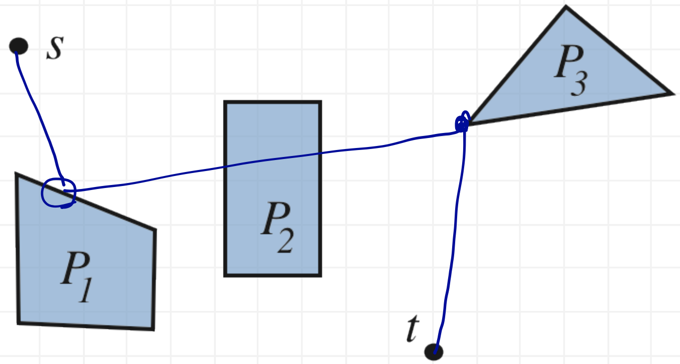
Touring a sequence of polygons
M Dror, A Efrat, A Lubiw, JSB Mitchell - … of the thirty-fifth annual ACM …, 2003 - dl.acm.org
Abstract Given a **sequence** of k **polygons** in the plane, a start point s, and a target point, t, we seek a shortest path that starts at s, visits in order each of the **polygons**, and ends at t. If the **polygons** are disjoint and convex, we give an algorithm running in time O (kn log (n/k)), **...**
Cited by 85Related articlesAll 11 versionsCiteSave

From: http://scholar.google.ca/scholar?hl=en&q=touring+a+sequence+of+polygons&btnG=&as_sdt=1%2C5&as_sdtp=

Problem: Given a sequence of k polygons in the plane, a start point s, and a target point, t, find a shortest path that starts at s, visits the polygons in order, and ends at t.

Algorithm:
$O(nk \log n)$ for disjoint convex polygons. n = total number of vertices.
$O(nk^2 \log n)$ for non-disjoint convex polygons, with boundary constraint on the path

Query for shortest path to t in $O(k \log n$ + output-size)



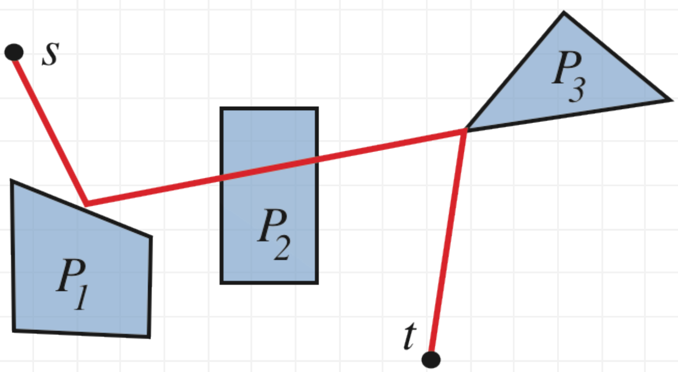
Touring a sequence of polygons

Touring a sequence of polygons
M Dror, A Efrat, A Lubiw, JSB Mitchell - … of the thirty-fifth annual ACM …, 2003 - dl.acm.org
Abstract Given a **sequence** of k **polygons** in the plane, a start point s, and a target point, t, we seek a shortest path that starts at s, visits in order each of the **polygons**, and ends at t. If the **polygons** are disjoint and convex, we give an algorithm running in time O (kn log (n/k)), **...**

Problem: Given a sequence of k polygons in the plane, a start point s, and a target point, t, find a shortest path that starts at s, visits the polygons in order, and ends at t.

Algorithm:
$O(nk \log n)$ for disjoint convex polygons. n = total number of vertices.
$O(nk^2 \log n)$ for non-disjoint convex polygons, with boundary constraint on the path

Query for shortest path to t in $O(k \log n$ + output-size)

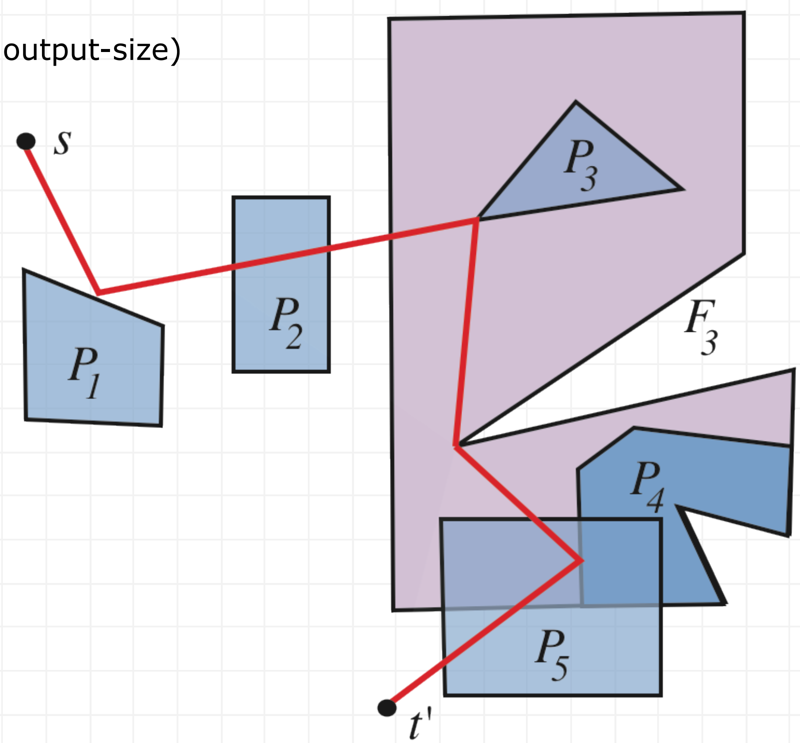


Touring a sequence of polygons

Problem: Given a sequence of k polygons in the plane, a start point s, and a target point, t, find a shortest path that starts at s, visits the polygons in order, and ends at t.

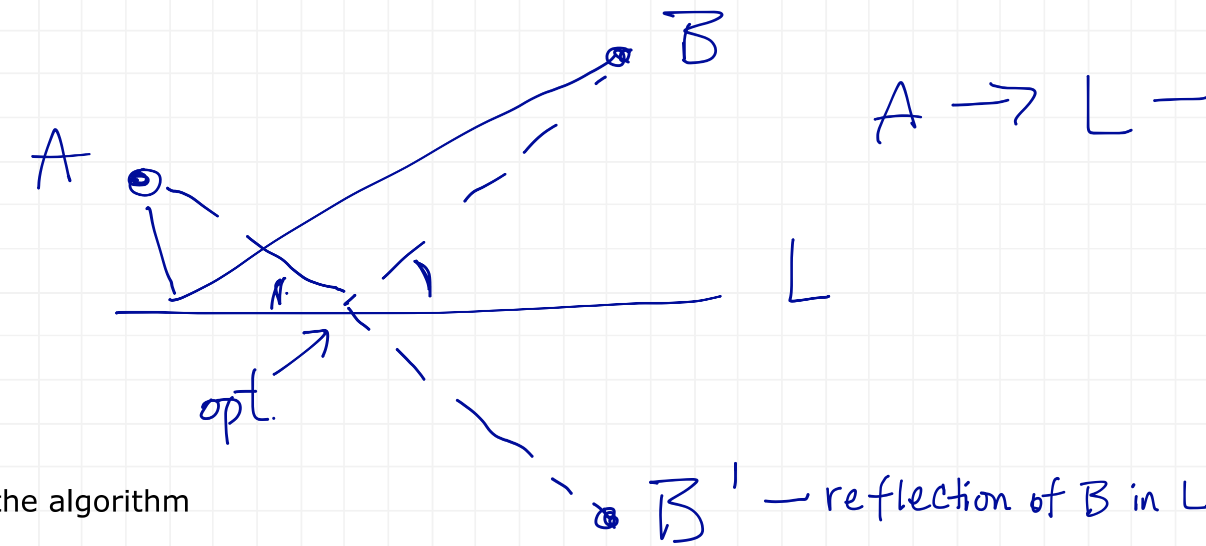
Algorithm:
$O(nk \log n)$ for disjoint convex polygons. n = total number of vertices.
$O(nk^2 \log n)$ for non-disjoint convex polygons, with boundary constraint on the path

Query for shortest path to t in $O(k \log n$ + output-size)



Touring a sequence of polygons

crucial property — visiting a line

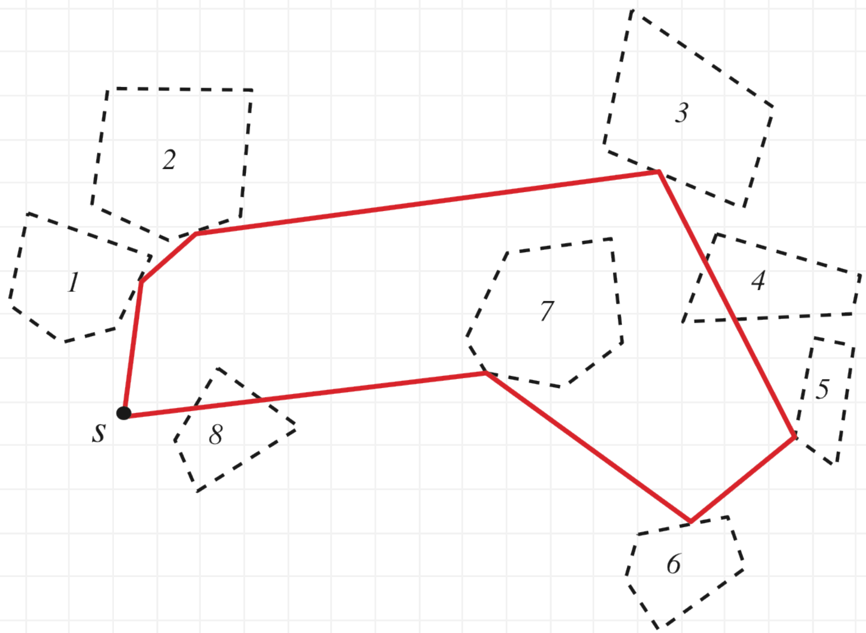


two main ideas for the algorithm

- locally shortest paths are unique
- although the shortest path map is too big, we can use a “last step shortest path map”

Touring a sequence of polygons — Applications

Parts Cutting



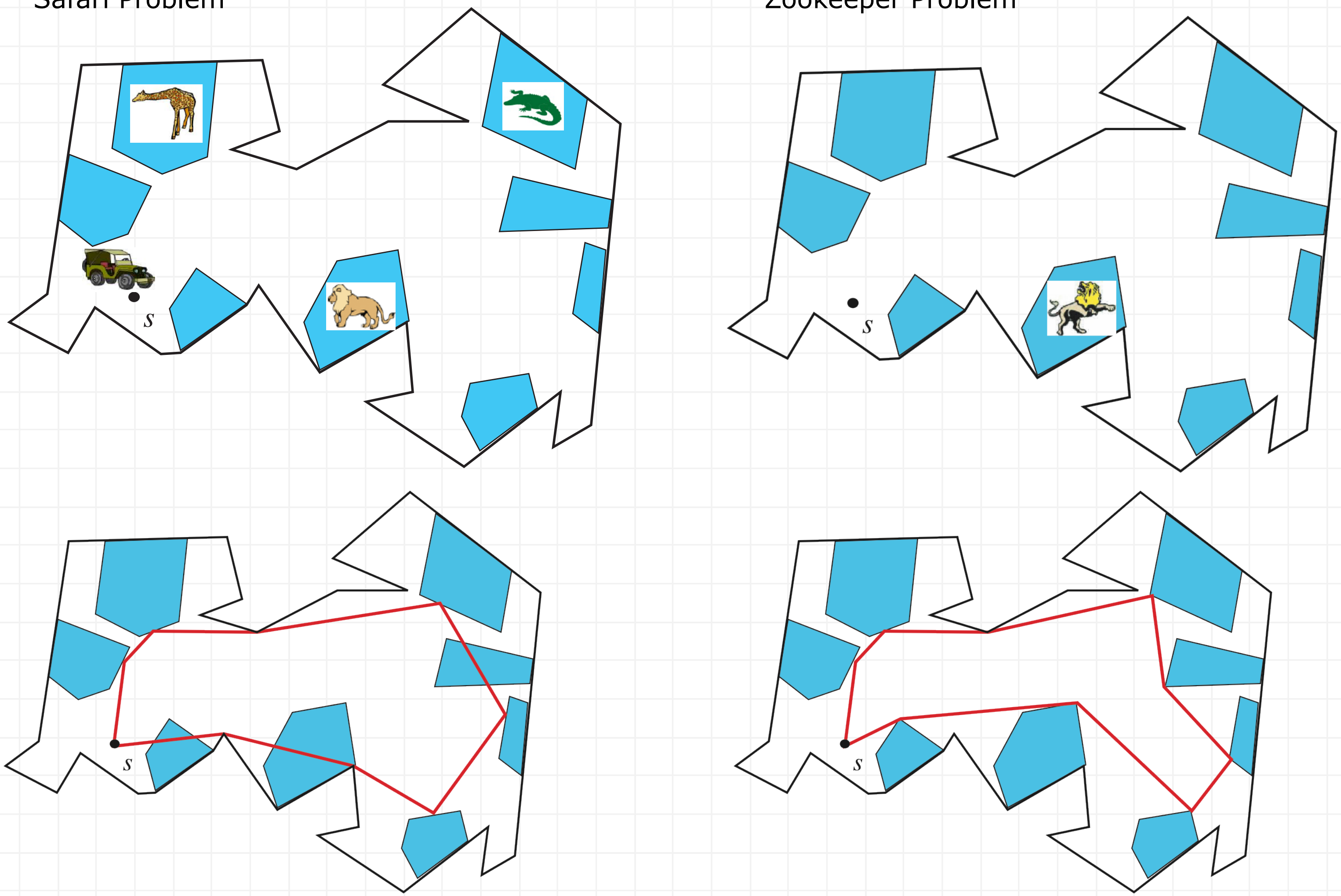
disjoint convex polygons (no boundaries)

O(nk log n) k = # polygons, n = total number of vertices

Touring a sequence of polygons — Applications

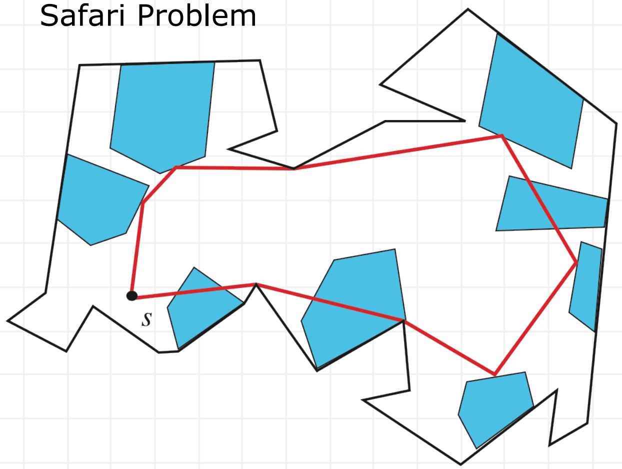
Safari Problem

Zookeeper Problem

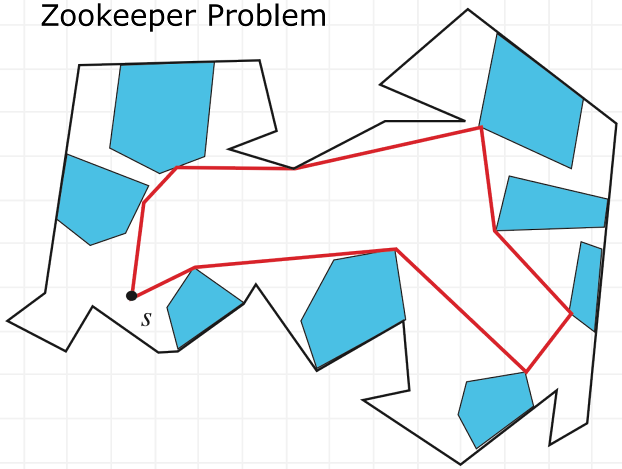


Touring a sequence of polygons — Applications

Safari Problem

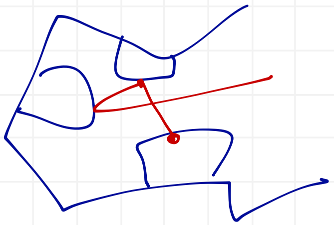


Zookeeper Problem



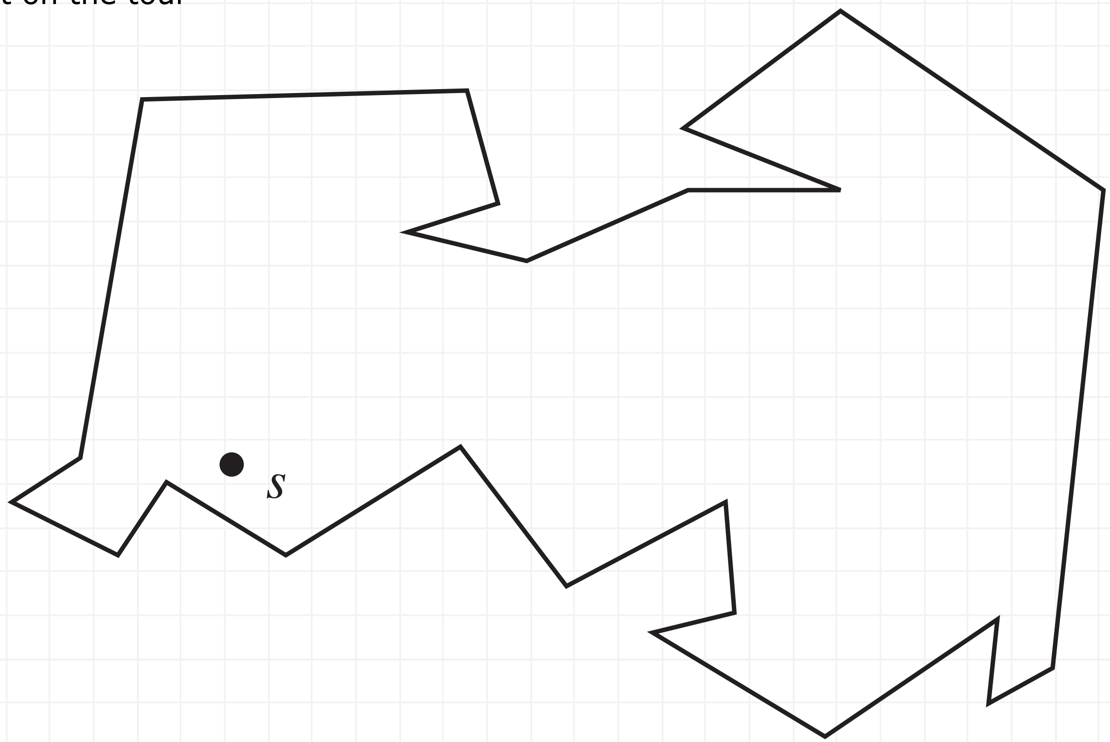
disjoint convex polygons
enclosed in one large polygon, and touching its boundary
order of visit determined by order around polygon — to prove

$O(nk \log n)$
this improved previous algorithms for Safari problem
(zookeeper has an $O(n \log n)$ algorithm)



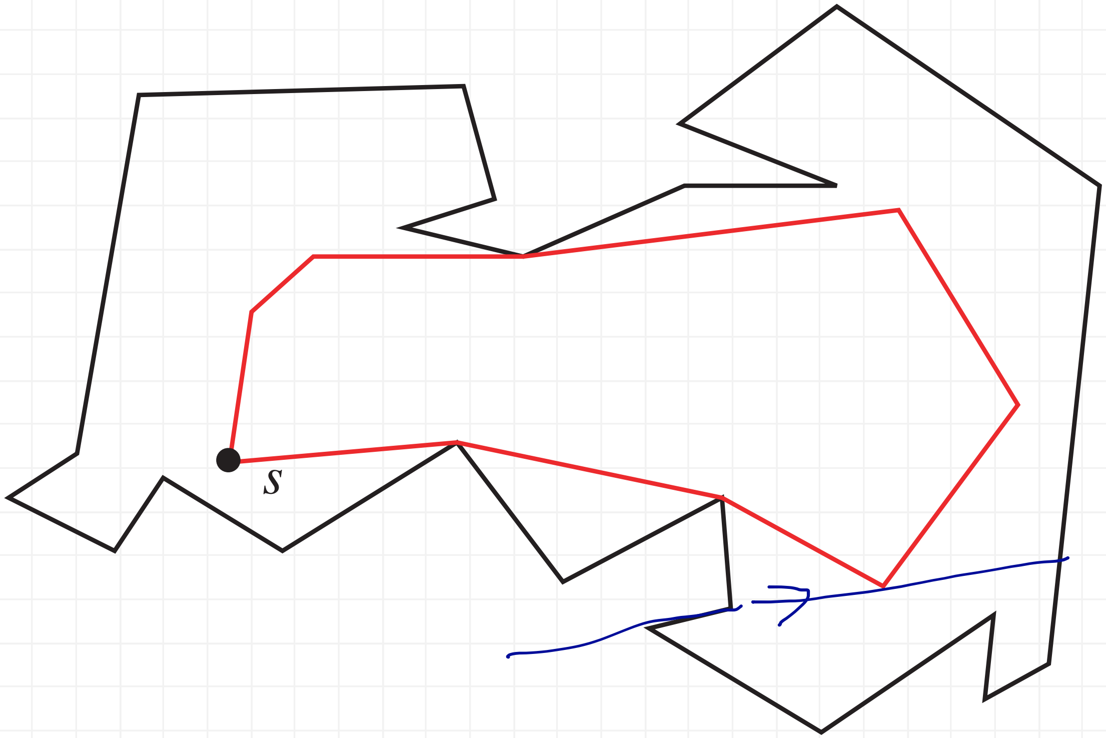
Touring a sequence of polygons — Applications

Watchman route — find shortest tour from *s* s.t. every point on the polygon boundary is visible from some point on the tour



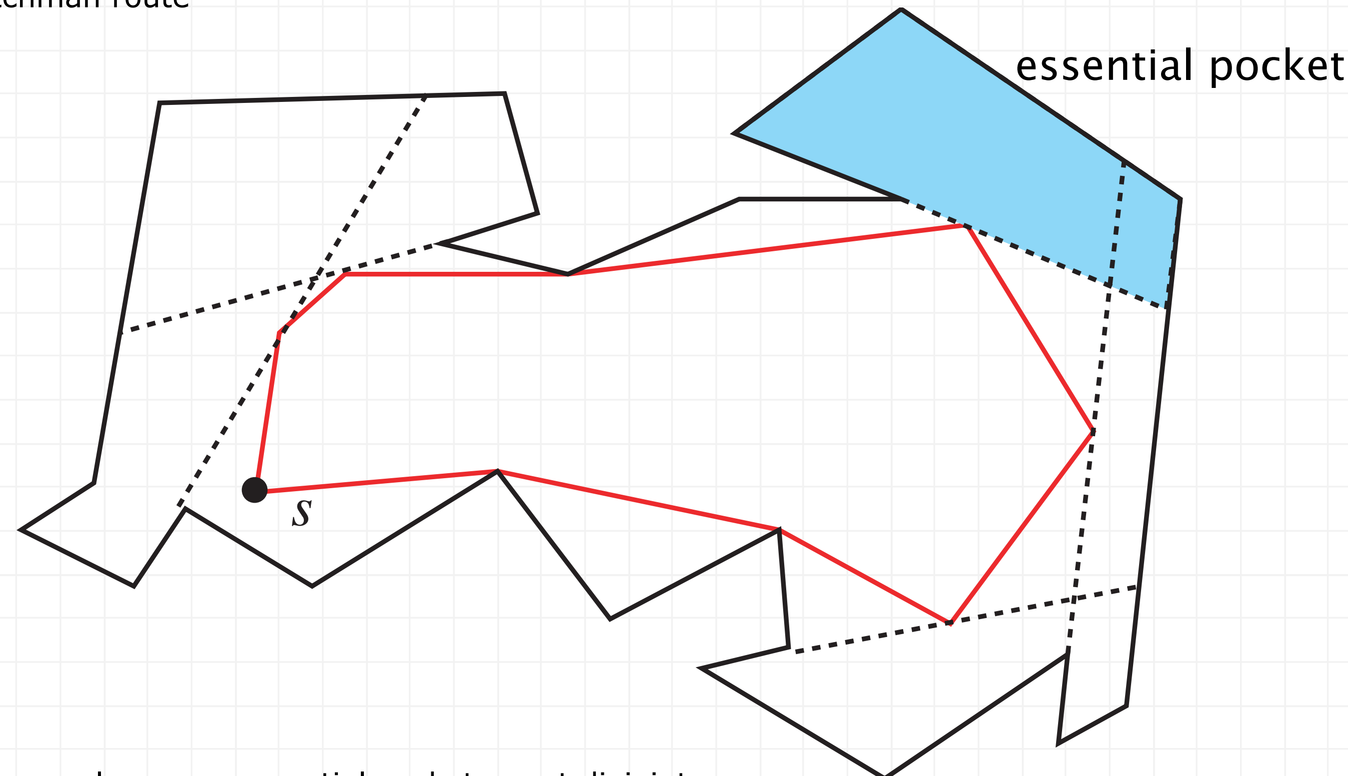
Touring a sequence of polygons — Applications

Watchman route — find shortest tour from *s* s.t. every point on the polygon boundary is visible from some point on the tour



Touring a sequence of polygons — Applications

Watchman route



convex polygons = essential pockets. not disjoint
enclosed in one large polygon, and touching its boundary
order of visit determined by order around polygon

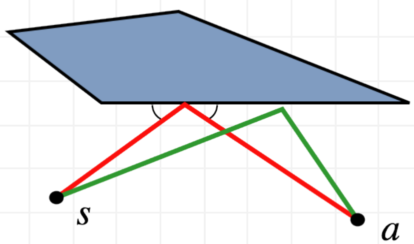
O(n^3 log n) — better than previous algorithms

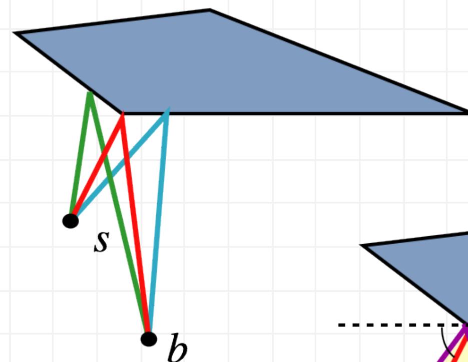
Touring a sequence of polygons

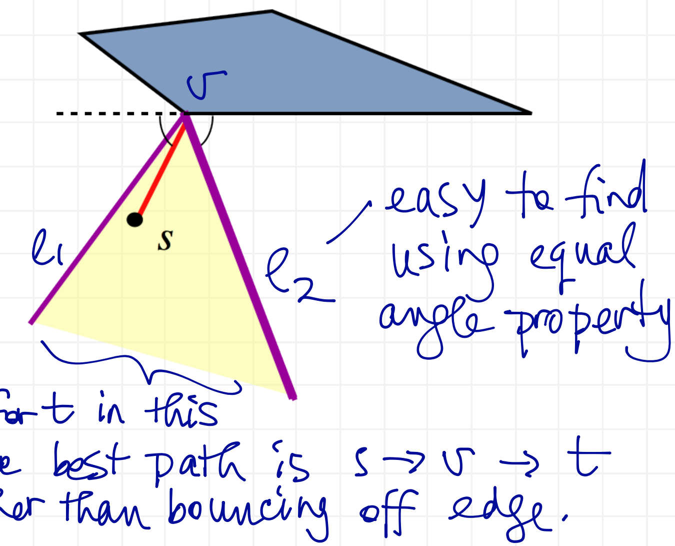
Two main ideas for the algorithm

- locally shortest paths are unique
- although the shortest path map is too big, we can use a "last step shortest path map"

A path is locally shortest if moving any one bend of the path does not improve it







Locally shortest ⟹ globally shortest

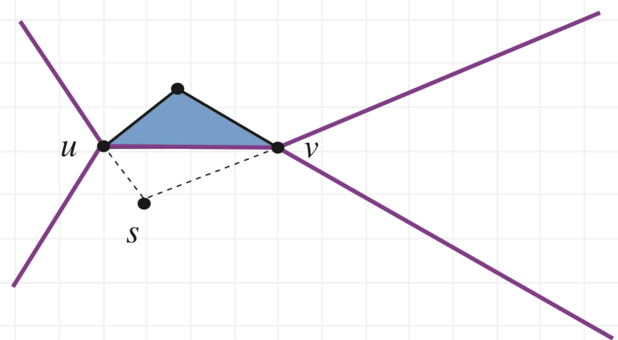
because locally shortest paths are unique
(as the algorithm will show)

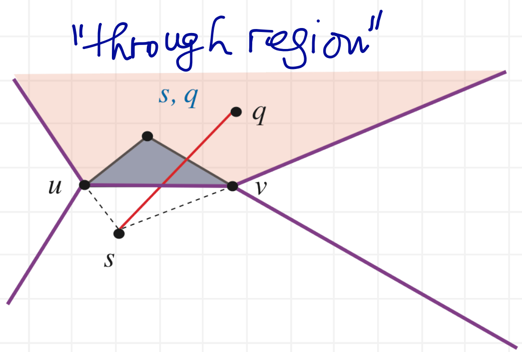
Touring a sequence of polygons

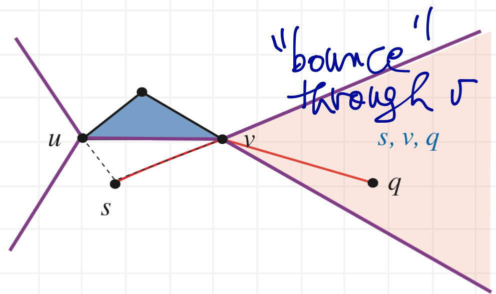
Two main ideas for the algorithm

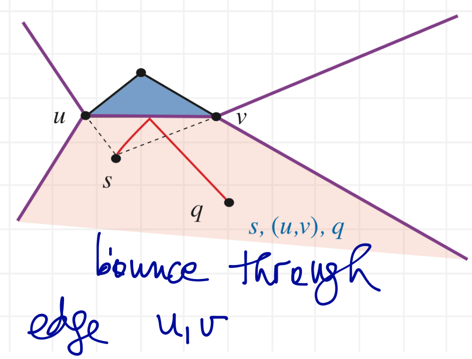
- locally shortest paths are unique
- although the shortest path map is too big, we can use a “last step shortest path map”

shortest path map — divide plane into regions by combinatorics of shortest path



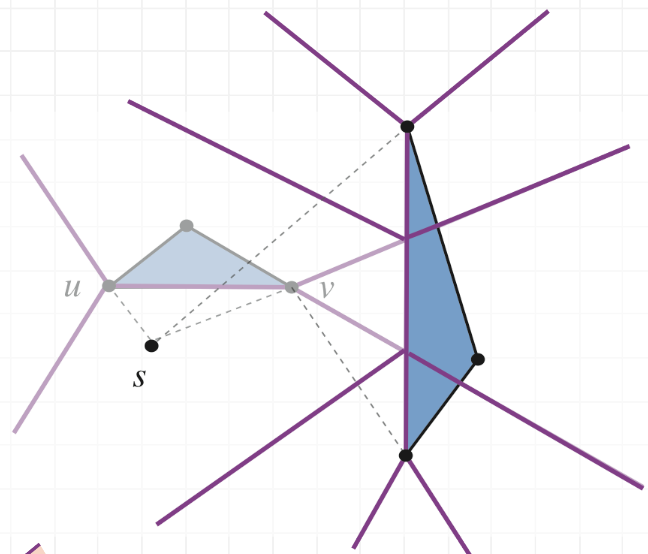




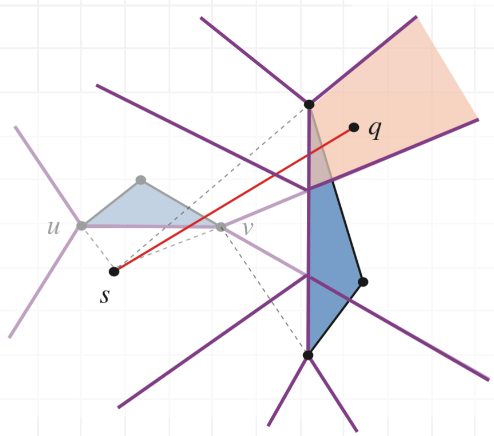


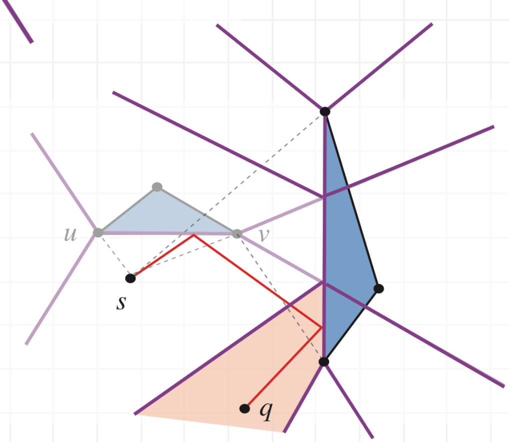
Touring a sequence of polygons

shortest path map — divide plane into regions by combinatorics of shortest path



4 regions for P_1
× 4 regions for P_2
= 16 — upper bound
(in fact have fewer)

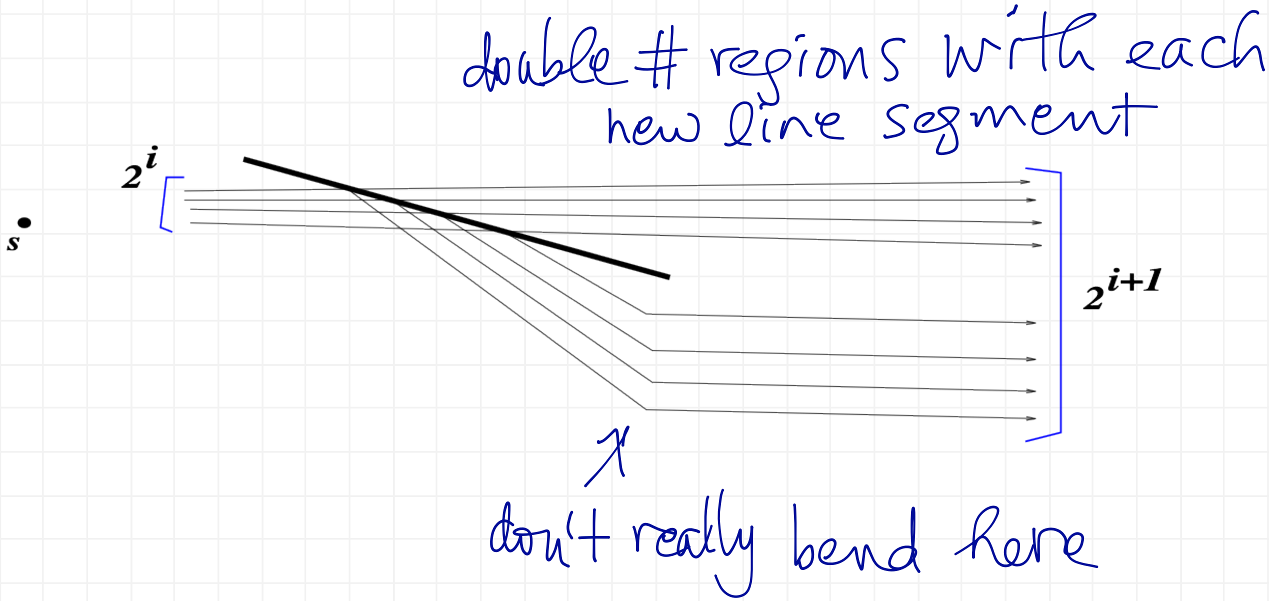




Touring a sequence of polygons

shortest path map — divide plane into regions by combinatorics of shortest path

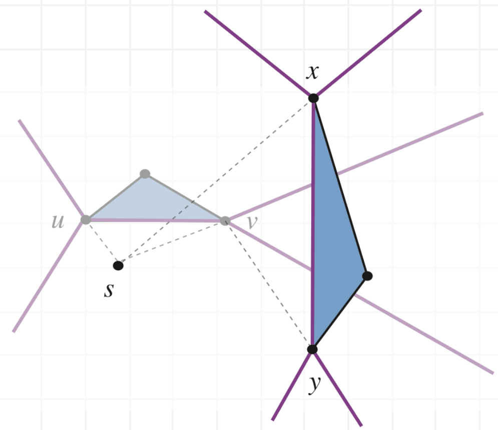
Theorem. The complexity of the shortest path map for n line segments can be 2^k

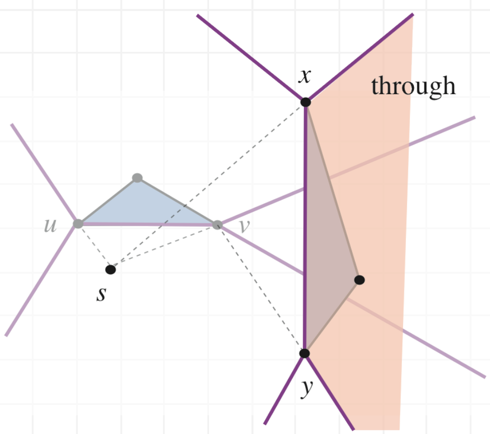


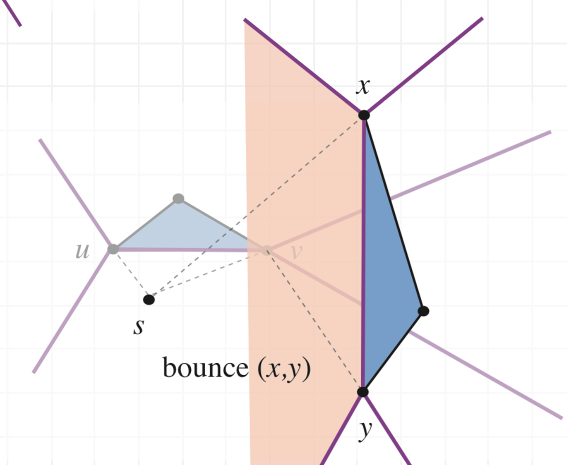
Touring a sequence of polygons

shortest path map — divide plane into regions by combinatorics of shortest path

last step shortest path map — divide plane into regions by combinatorics of the last step of the shortest path



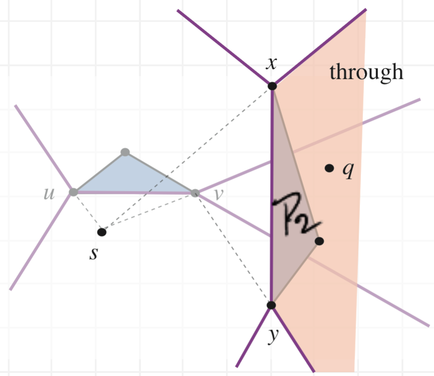




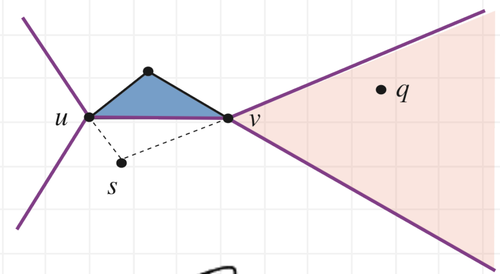
Touring a sequence of polygons

last step shortest path map — divide plane into regions by combinatorics of the last step of the shortest path

answering queries using the last step shortest path map — Example 1

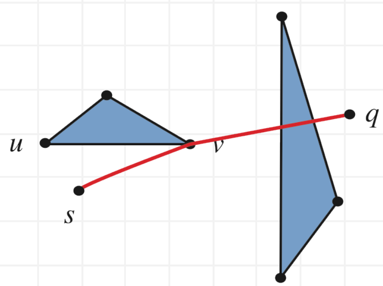


query P_2



query P_1

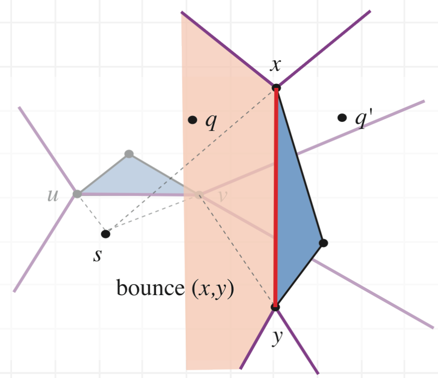
final result



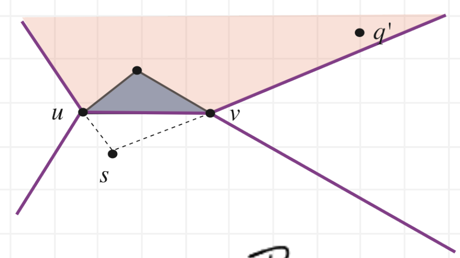
Touring a sequence of polygons

last step shortest path map — divide plane into regions by combinatorics of the last step of the shortest path

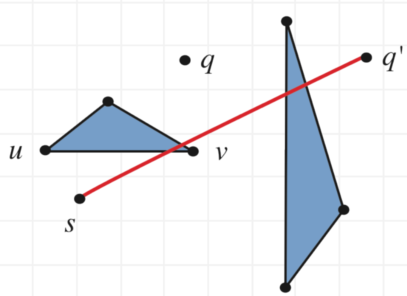
answering queries using the last step shortest path map — Example 2



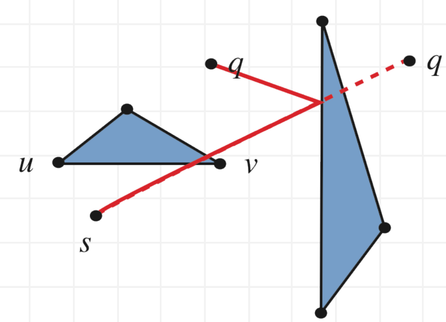
query P_2



query P_1



final result



Example 3 – bounce off vertex y for P_2 ⟹ query y for P_1

Touring a sequence of polygons

last step shortest path map — divide plane into regions by combinatorics of the last step of the shortest path

answering queries using the last step shortest path map

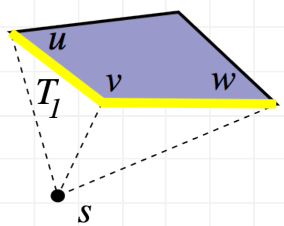
O(k log n) to find shortest path length
O(k log n + output-size) to find shortest path

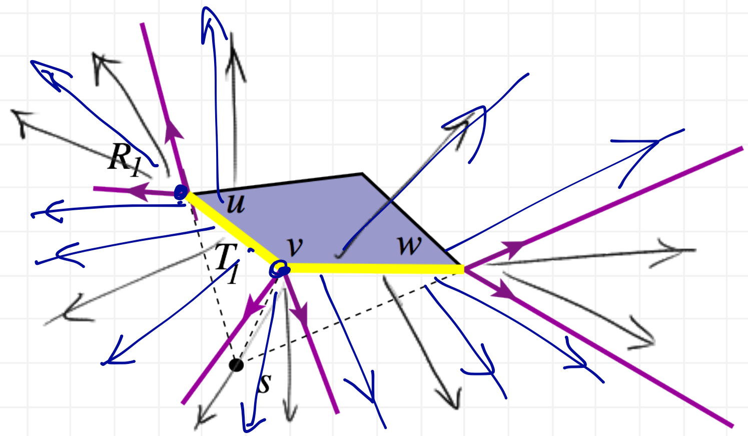
need:
- planar point location
- reflect point in line

don't need distance computations to find shortest path (so no square roots)

Touring a sequence of polygons — Algoirithm

Add polygons one by one, computing the last step shortest path map for each





find T = “first contact” points of P

find the rays leaving vertices of T

Claim. T forms a chain.

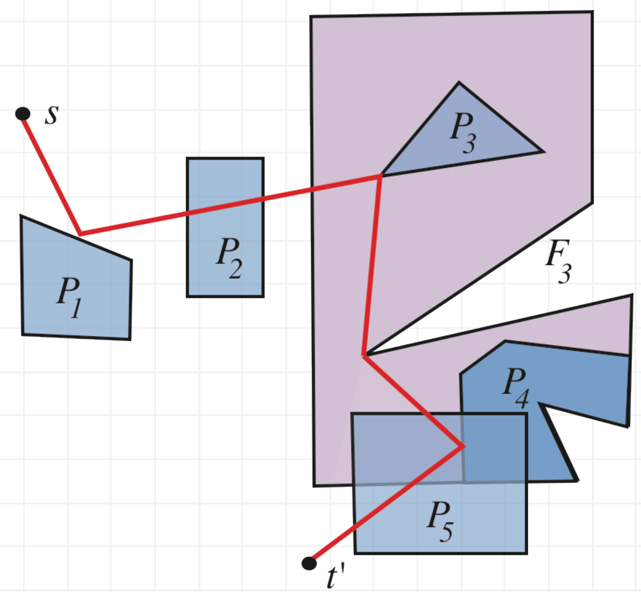
Claim. rays leaving T form a “starburst”, i.e. there is a unique ray to each point in the plane

Thus locally shortest paths are unique

Just need to query every vertex of P_i with respect to polygon sequence $P_1 \ldots P_{i-1}$
O(k log n) per query. Total O(nk log n).

Touring a sequence of polygons

general problem: non-disjoint convex polygons, with boundary constraints on the path

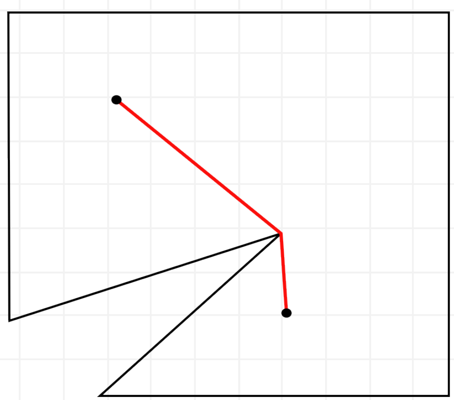


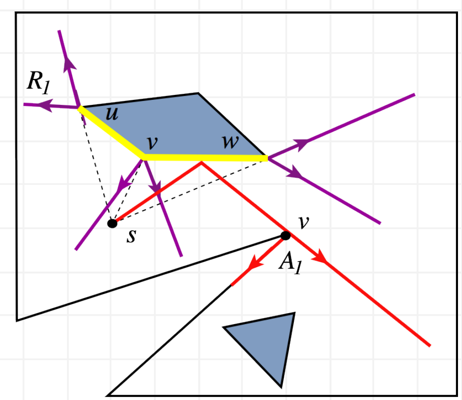
algorithm and analysis are more complicated — will just give top-level ideas

Touring a sequence of polygons

general problem: non-disjoint convex polygons, with boundary constraints on the path

boundary constraints



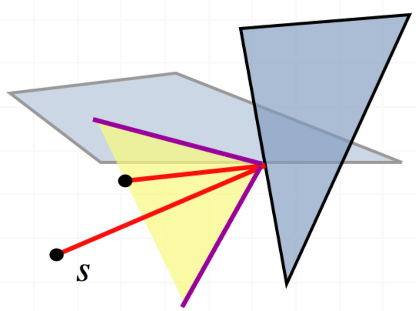


construct shortest path map from the starburst rays leaving *P*

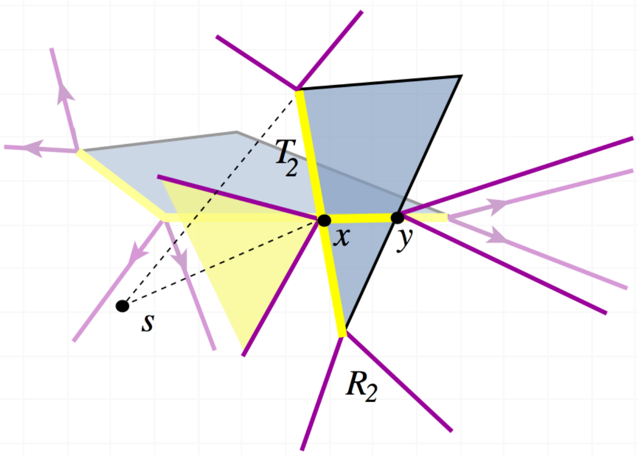
Touring a sequence of polygons

general problem: non-disjoint convex polygons, with boundary constraints on the path

intersecting polygons



shortest paths may “bounce” from intersection points



“first contact” set becomes a tree
rays leaving P still form a starburst
so locally shortest paths are unique

Running time increases to $O(nk^2 \log n)$

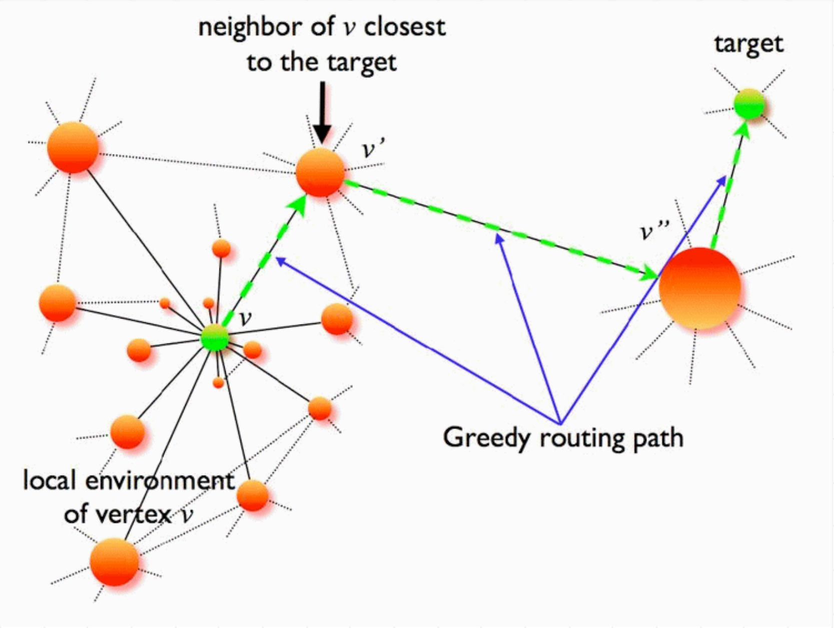
routing

Find a path from s to t in a [geometric] graph without knowing the graph.

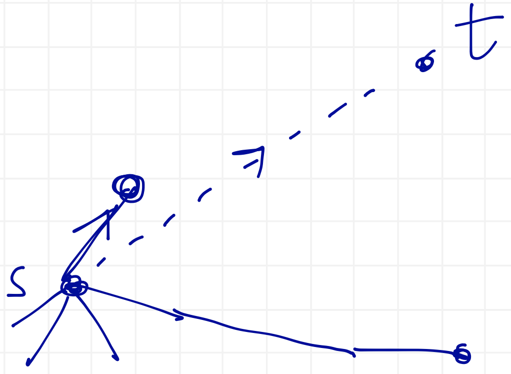
Use local information.

geometric routing — graph G embedded in the plane

Greedy routing



compass routing

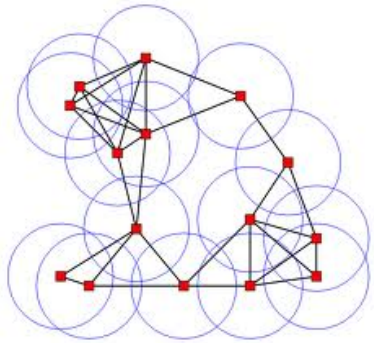


geometric routing

Routing with guaranteed delivery in ad hoc wireless networks
P Bose, P Morin, I Stojmenović, J Urrutia - **Wireless networks**, 2001 - dl.acm.org
... In contrast, our algorithms always guarantee that a packet will be **delivered** to (all of) its intended recipient(s) so long as the unit graph U(S) is static and connected during the time it takes to route a message. **...** Page 3. **ROUTING** WITH **GUARANTEED DELIVERY** IN **AD HOC** **...**
Cited by 1833Related articlesAll 52 versionsCiteSave

From: http://scholar.google.ca/scholar?hl=en&q=Routing+with+guar-+anteed+delivery+in+ad+hoc+wireless+networks&btnG=&as_sdt=1%2C5&as_sdtp=

in unit disk graph.



Online routing in triangulations
P Bose, P Morin - SIAM journal on computing, 2004 - SIAM
We consider **online routing** algorithms for **routing** between the vertices of embedded planar straight line graphs. Our results include (1) two deterministic memoryless **routing** algorithms, one that works for all Delaunay **triangulations** and the other that works for all regular **...**
Cited by 87Related articlesAll 2 versionsCiteSave

– planar triangulations

From: http://scholar.google.ca/scholar?hl=en&q=Online+routing+in+triangulations&btnG=&as_sdt=1%2C5&as_sdtp=

routing

for an abstract graph, we can embed it using “virtual coordinates” and use geometric routing

[HTML] **On a conjecture related to geometric routing**
CH Papadimitriou, D Ratajczak - Theoretical Computer Science, 2005 - Elsevier
We **conjecture** that any planar 3-connected graph can be embedded in the plane in such a way that for any nodes s and t, there is a path from s to t such that the Euclidean distance to t decreases monotonically along the path. A consequence of this **conjecture** would be that **...**
Cited by 107 Related articles All 7 versions Cite Save

From: http://scholar.google.ca/scholar?hl=en&q=On+a+conjecture+related+to+geometric+routing&btnG=&as_sdt=1%2C5&as_sdtp=

— conj: any 3-connected graph can be embedded s.t. greedy routing works

Some results on greedy embeddings in metric spaces
T Leighton, A Moitra - Discrete & Computational Geometry, 2010 - Springer
Abstract Geographic Routing is a family of routing algorithms that uses geographic point locations as addresses for the purposes of routing. Such routing algorithms have proven to be both simple to implement and heuristically effective when applied to wireless sensor **...**
Cited by 70 Related articles All 21 versions Cite Save

From: http://scholar.google.ca/scholar?hl=en&q=Some+Results+on+Greedy+Embeddings+in+Metric+Spaces&btnG=&as_sdt=1%2C5&as_sdtp=

— positive results but # bits for coords. too high (routing tables better)

Succinct greedy drawings do not always exist
P Angelini, G Di Battista, F Frati - Networks, 2012 - Wiley Online Library
Abstract A **greedy drawing** is a graph **drawing** containing a distance-decreasing path for every pair of nodes. A path (v 0, v 1,…, vm) is distance-decreasing if d (vi, vm)< d (v i-1, vm), for i= 1,…, m. **Greedy drawings** easily support geographic **greedy** routing. Hence, a **...**
Cited by 3 Related articles All 3 versions Cite Save

From: http://scholar.google.ca/scholar?hl=en&q=Succinct+greedy+drawings+do+not+always+exist&btnG=&as_sdt=1%2C5&as_sdtp=

— for trees

competitive online routing

— want length of path found using local routing
$\leq c \cdot$ shortest path

[HTML] **Competitive online routing in geometric graphs**
P Bose, P Morin - Theoretical Computer Science, 2004 - Elsevier
We consider **online routing** algorithms for finding paths between the vertices of plane **graphs**. Although it has been shown in Bose et al.(Internat. J. Comput. Geom. 12 (4)(2002) 283) that there exists no **competitive routing** scheme that works on all triangulations, we **...**
Cited by 59 Related articles All 34 versions Cite Save

From: http://scholar.google.ca/scholar?q=Competitive+online+routing+in+geometric+graphs&btnG=&hl=en&as_sdt=0%2C5

— + more papers (see web page)